The Keys to Decidable HyperLTL Satisfiability: Small Models or Very Simple Formulas

Corto Macle[1] and Martin Zimmermann[2]

[1] ENS Paris-Saclay, Cachan, France
corto.mascle@ens-paris-saclay.fr
[2] University of Liverpool, Liverpool, United Kingdom
martin.zimmermann@liverpool.ac.uk

Abstract. HyperLTL, the extension of Linear Temporal Logic by trace quantifiers, is a uniform framework for expressing information flow policies by relating multiple traces of a security-critical system. HyperLTL has been successfully applied to express fundamental security policies like noninterference and observational determinism, but has also found applications beyond security, e.g., distributed protocols and coding theory. However, HyperLTL satisfiability is undecidable as soon as there are existential quantifiers in the scope of a universal one. To overcome this severe limitation to applicability, we investigate here restricted variants of the satisfiability problem to pinpoint the decidability border.
First, we restrict the space of admissible models and show decidability when restricting the search space to models of bounded size or to finitely representable ones. Second, we consider formulas with restricted nesting of temporal operators and show that nesting depth one yields decidability for a slightly larger class of quantifier prefixes. We provide tight complexity bounds in almost all cases.

1 Introduction

The introduction of temporal logics for the specification of information flow policies [2] was a significant milestone in the long and successful history of applying logics in computer science [14]. Probably the most important representative of these logics is HyperLTL [2], which extends Linear Temporal Logic (LTL) [21] by trace quantifiers. This addition allows to express properties that relate multiple execution traces, which is typically necessary to capture the flow of information [3]. In contrast, LTL, currently the most influential specification language for reactive systems, is only able to express properties of single traces.

HyperLTL provides a uniform framework for expressing information flow policies in a formalism with intuitive syntax and semantics, and for the automated verification of these policies: A wide range of policies from the literature [13,17,18,19,20,25] with specialized verification algorithms is expressible in HyperLTL, i.e., universal HyperLTL verification algorithms are applicable to all of them.

As an example, consider a system with a set I of inputs, which contains a hidden input $h \in I$, and an output o. Now, noninterference [13] between h and o requires that no information about h is leaked via o, i.e., for all execution traces π and π', if the inputs in π and π' only differ in h, then they have the same output at all times. Formally, this is captured by the HyperLTL formula

$$\forall \pi. \forall \pi'. \left(\mathbf{G} \bigwedge\nolimits_{i \in I \setminus \{h\}} (i_\pi \Leftrightarrow i_{\pi'}) \right) \Rightarrow \mathbf{G}\,(o_\pi \Leftrightarrow o_{\pi'}).$$

Today, there are tools for model checking HyperLTL properties [5,11], for checking satisfiability of HyperLTL properties [9], for synthesizing reactive systems from HyperLTL properties [8], and for runtime monitoring of HyperLTL properties [1,10]. Furthermore, the extraordinary expressiveness of HyperLTL has been exhibited [12] and connections to first and second-order predicate logics have been established [4,12].

The major drawback of HyperLTL is the usual price one has to pay for great expressiveness: prohibitively high worst-case complexity. In particular, model checking finite Kripke structures against HyperLTL formulas is nonelementary [2] and satisfiability is even undecidable [7]. These results have to be contrasted with model checking and satisfiability being PSpace-complete for LTL [23], problems routinely solved in real-life applications [16].

Due to the sobering state of affairs, it is imperative to find fragments of the logic with (more) tractable complexity. In this work, we focus on the satisfiability problem, the most fundamental decision problem for a logic. Nevertheless, it has many applications in verification, e.g., checking the equivalence and implication of specifications can be reduced to satisfiability. Finally, the question whether a property given by some HyperLTL formula is realizable by some system is also a satisfiability problem.

A classical attempt to overcome the undecidability of the satisfiability problem is to restrict the number of quantifier alternations of the formulas under consideration. In fact, the alternation depth is the measure underlying the nonelementary complexity of the HyperLTL model checking problem [2]. However, the situation is different for the satisfiability problem: It is undecidable even when restricted to $\forall\exists^*$ formulas, i.e., formulas starting with one universal quantifier followed by existential ones [7]. All remaining prefix classes are decidable by reductions to the LTL satisfiability problem, e.g., the satisfiability problem is PSpace-complete for the alternation-free prefix classes $\exists^*$ and $\forall^*$ and ExpSpace-complete for the class $\exists^*\forall^*$ [7].

However, there are more complexity measures beyond the alternation depth that can be restricted in order to obtain tractable satisfiability problems, both on formulas and on models. The latter case is of particular interest, since it is known that not every satisfiable HyperLTL has a "simple" model, for various formalizations of "simple" [12]. Thus, for those formulas, such a restriction could make a significant difference. Furthermore, from a more practical point of view, one is often interested in whether there is a, say, finite model while the existence of an intricate infinite model may not be useful.

We study the satisfiability problem for formulas with restricted quantifier prefixes and restricted temporal depth [6], which measures the nesting of temporal operators. Our main result here shows that satisfiability is even undecidable for formulas of the form $\forall^2\exists^*\varphi$, where φ has temporal depth one and only uses eventually $\mathbf{F}$ and always $\mathbf{G}$, i.e., it is a Boolean combination of formulas $\mathbf{F}\,\varphi'$ with propositional φ'. Thereby, we strengthen the previous undecidability result for $\forall\exists^*$ by bounding the temporal depth to one, but at the price of a second universal quantifier. Moreover, we clarify the border between decidability and undecidability at temporal depth two: Using only one universally quantified variable, temporal depth one, and only $\mathbf{F}$, $\mathbf{G}$, and nested applications of next $\mathbf{X}$ leads to decidability. Finally, we show that every HyperLTL formula can be transformed into an equisatisfiable $\forall^2\exists^*$ formula of temporal depth two, i.e., this fragment already captures the full complexity of the satisfiability problem.

Thus, the overall picture is still rather bleak: if one only restricts the formula then the islands of decidability are very small. Phrased differently, even very simple formulas are extremely expressive and allow to encode computations of Turing-complete devices in their models. However, note that such models are necessarily complex, as they need to be able to encode an unbounded amount of information.

Thus, we also consider satisfiability problems for arbitrary formulas, but with respect to restricted models which do not allow to encode such computations. In particular, we consider three variants of increasing complexity: Checking whether a given HyperLTL formula has a model of a given cardinality k is ExpSpace-complete, whether it has a model containing only ultimately periodic traces of length at most k is N2ExpTime-complete, and checking whether it has a model induced by a Kripke structure with k states is Tower-complete. The last result is even true for a fixed Kripke structure, which therefore has implications for the complexity of the model checking problem as well. Thus, the situation is more encouraging when checking for the existence of small models: satisfiability becomes decidable, even with (relatively) moderate complexity in the first two cases.

However, as argued above, all three approaches are (necessarily) incomplete: There are satisfiable formulas that have only infinite models, satisfiable formulas that have only non-ultimately periodic models, and satisfiable formulas that have no ω-regular models [12], a class of models that includes all those that are induced by a finite Kripke structure.

All in all, our work shows that HyperLTL satisfiability remains a challenging problem, but we have provided a complete classification of the tractable cases in terms of alternation depth, temporal depth, and representation of the model (for formulas without until).

All proofs omitted due to space restrictions can be found in the appendix.

2 Definitions

Fix a finite set AP of atomic propositions. A *valuation* is a subset of AP. A *trace* over AP is a map $t\colon \mathbb{N} \to 2^{\mathrm{AP}}$, denoted by $t(0)t(1)t(2)\cdots$, i.e., an infinite sequence of valuations. The set of all traces over

AP is denoted by $(2^{\mathrm{AP}})^\omega$. The *projection* of t to AP' is the trace $(t(0)\cap \mathrm{AP}')(t(1)\cap \mathrm{AP}')(t(2)\cap \mathrm{AP}')\cdots$ over AP'. A trace t is *ultimately periodic*, if $t = x\cdot y^\omega$ for some $x, y \in (2^{\mathrm{AP}})^+$, i.e., there are $s, p > 0$ with $t(n) = t(n+p)$ for all $n \geq s$.

The formulas of HyperLTL are given by the grammar

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$$
$$\psi ::= a_\pi \mid \neg\psi \mid \psi\vee\psi \mid \mathbf{X}\,\psi \mid \psi\,\mathbf{U}\,\psi$$

where a ranges over atomic propositions in AP and where π ranges over a fixed countable set $\mathcal{V}$ of *trace variables*. Conjunction, implication, equivalence, and exclusive disjunction $\oplus$, as well as the temporal operators eventually $\mathbf{F}$ and always $\mathbf{G}$ are derived as usual. A *sentence* is a closed formula, i.e., a formula without free trace variables. The *size* of a formula φ, denoted by $|\varphi|$, is its number of distinct subformulas.

The semantics of HyperLTL is defined with respect to a *trace assignment*, a partial mapping $\Pi\colon \mathcal{V} \to (2^{\mathrm{AP}})^\omega$. The assignment with empty domain is denoted by $\Pi_\emptyset$. Given a trace assignment Π, a trace variable π, and a trace t we denote by $\Pi[\pi\to t]$ the assignment that coincides with Π everywhere but at π, which is mapped to t. We also use shorthand notation like $[\pi_1\to t_1,\ldots,\pi_n\to t_n]$ and $[(\pi_i\to t_i)_{1\leq i\leq n}]$ for $\Pi_\emptyset[\pi_1\to t_1]\ldots[\pi_n\to t_n]$, if the π_i are pairwise different. Furthermore, $\Pi[j,\infty)$ denotes the trace assignment mapping every π in Π's domain to $\Pi(\pi)(j)\Pi(\pi)(j+1)\Pi(\pi)(j+2)\cdots$.

For sets T of traces and trace assignments Π we define

- $(T,\Pi)\models a_\pi$, if $a\in\Pi(\pi)(0)$,
- $(T,\Pi)\models\neg\psi$, if $(T,\Pi)\not\models\psi$,
- $(T,\Pi)\models\psi_1\vee\psi_2$, if $(T,\Pi)\models\psi_1$ or $(T,\Pi)\models\psi_2$,
- $(T,\Pi)\models\mathbf{X}\,\psi$, if $(T,\Pi[1,\infty))\models\psi$,
- $(T,\Pi)\models\psi_1\,\mathbf{U}\,\psi_2$, if there is a $j\geq 0$ such that $(T,\Pi[j,\infty))\models\psi_2$ and for all $0\leq j'<j$: $(T,\Pi[j',\infty))\models\psi_1$,
- $(T,\Pi)\models\exists\pi.\varphi$, if there is a trace $t\in T$ such that $(T,\Pi[\pi\to t])\models\varphi$, and
- $(T,\Pi)\models\forall\pi.\varphi$, if for all traces $t\in T$: $(T,\Pi[\pi\to t])\models\varphi$.

We say that T *satisfies* a sentence φ if $(T,\Pi_\emptyset)\models\varphi$. In this case, we write $T\models\varphi$ and say that T is a *model* of φ. Conversely, satisfaction of quantifier-free formulas does not depend on T. Hence, we say that Π *satisfies* a quantifier-free ψ if $(\emptyset,\Pi)\models\psi$ and write $\Pi\models\psi$ (assuming Π is defined on all trace variables that appear in ψ).

The *alternation depth* of a HyperLTL formula φ, denoted by $\mathrm{ad}(\varphi)$, is defined as its number of quantifier alternations. Its *temporal depth*, denoted by $\mathrm{td}(\varphi)$, is defined as the maximal depth of the nesting of temporal operators in the formula. Formally, td and ad are defined as follows :

- $\mathrm{td}(a_\pi)=0$
- $\mathrm{td}(\neg\psi)=\mathrm{td}(\psi)$
- $\mathrm{td}(\psi_1\vee\psi_2)=\max(\mathrm{td}(\psi_1),\mathrm{td}(\psi_2))$,
- $\mathrm{td}(\mathbf{X}\,\psi)=1+\mathrm{td}(\psi)$,
- $\mathrm{td}(\psi_1\,\mathbf{U}\,\psi_2)=1+\max(\mathrm{td}(\psi_1),\mathrm{td}(\psi_2))$,
- $\mathrm{td}(\exists\pi.\varphi)=\mathrm{td}(\varphi)$
- $\mathrm{td}(\forall\pi.\varphi)=\mathrm{td}(\varphi)$.

- $\mathrm{ad}(\exists\pi_1\ldots\exists\pi_n.\varphi)=0$ for quantifier-free φ
- $\mathrm{ad}(\forall\pi_1\ldots\forall\pi_n.\varphi)=0$ for quantifier-free φ
- $\mathrm{ad}(\exists\pi_1\ldots\exists\pi_n.\forall\tau.\varphi)=1+\mathrm{ad}(\forall\tau.\varphi)$
- $\mathrm{ad}(\forall\pi_1\ldots\forall\pi_n.\exists\tau.\varphi)=1+\mathrm{ad}(\exists\tau.\varphi)$

Although HyperLTL sentences are required to be in prenex normal form, they are closed under Boolean combinations, which can easily be seen by transforming such formulas into prenex normal form. Note that this transformation can be implemented such that it changes neither the temporal nor alternation depth, and can be performed in polynomial time.

The fragment $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G})$ contains formulas of temporal depth one using only $\mathbf{F}$ and $\mathbf{G}$ as temporal operators, and $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G},\mathbf{X}^*)$ contains formulas using only $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{X}$ as temporal operators and of temporal depth one, however we allow iterations of the $\mathbf{X}$ operator. Formally, $\mathrm{HyperLTL}^1(\mathbf{F},\mathbf{G},\mathbf{X}^*)$ formulas are generated by the grammar

$$\varphi ::= \exists\pi.\varphi \mid \forall\pi.\varphi \mid \psi$$
$$\psi ::= \neg\psi \mid \psi\vee\psi \mid \psi\wedge\psi \mid \mathbf{X}^{\,n}\psi' \mid \mathbf{F}\,\psi' \mid \mathbf{G}\,\psi' \mid \psi'$$
$$\psi' ::= a_\pi \mid \neg\psi' \mid \psi'\vee\psi' \mid \psi'\wedge\psi'$$

where n ranges over the natural numbers. The grammar for HyperLTL$^1(\mathbf{F}, \mathbf{G})$ is obtained by removing $\mathbf{X}^n \psi'$ from the grammar above.

Also, we use standard notation for classes of formulas with restricted quantifier prefixes, e.g., $\forall^2\exists^*$ denotes the set of HyperLTL formulas in prenex normal form with two universal quantifiers followed by an arbitrary number of existential quantifiers, but no other quantifiers.

Finally, we encounter various complexity classes, classical ones from NP to N2ExpTime, as well as Tower (see, e.g., [22]). Intuitively, Tower is the set of problems that can be solved by a Turing machine that, on an input of size n, stops in time $2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}$, with the height of the tower of exponents bounded by $b(n)$, where b is a fixed elementary function. The reductions presented in this work are polynomial time reductions unless otherwise stated.

3 Satisfiability for Restricted Classes of Models

As the satisfiability problem "Given a HyperLTL sentence φ, does φ have a nonempty model?" is undecidable, even when restricted to finite models [7]. Hence, one has to consider simpler problems to regain decidability. In this section, we simplify the problem by checking only for the existence of *simple* models, for the following three formalizations of simplicity, where the bound k is always part of the input:

- Models of cardinality at most k (Theorem 1).
- Models containing only ultimately periodic traces xy^ω with $|x| + |y| \leq k$ (Theorem 2).
- Models induced by a finite-state system with at most k states (Theorem 3).

In every case, we allow arbitrary HyperLTL formulas as input and encode k in binary.

With the following result, we determine the complexity of checking satisfiability with respect to models of bounded size. The algorithm uses a technique introduced by Finkbeiner and Hahn [7, Theorem 3] that allows us to replace existential and universal quantification by disjunctions and conjunctions, if the model is finite. Similarly, the lower bound also follows from Finkbeiner and Hahn.

Theorem 1. *The following problem is* ExpSpace*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model with at most k traces?*

Proof. For the ExpSpace upper bound, one can check, given φ and k, satisfiability of the sentence $\exists\pi_1 \ldots \exists\pi_k.\overline{\varphi}$ where $\overline{\varphi}$ is defined inductively as follows:

- $\overline{\varphi} = \varphi$ if φ is quantifier-free.
- $\overline{\forall\pi.\varphi} = \bigwedge_{i=1}^{k} \overline{\varphi}[\pi \leftarrow \pi_i]$.
- $\overline{\exists\pi.\varphi} = \bigvee_{i=1}^{k} \overline{\varphi}[\pi \leftarrow \pi_i]$.

Here, $\overline{\varphi}[\pi \leftarrow \pi_i]$ is obtained from $\overline{\varphi}$ by replacing every occurrence of π by π_i. This sentence states the existence of at most k traces satisfying φ by replacing every quantifier by an explicit conjunction or disjunction over the possible assignments.

The resulting sentence is of size at most $|\varphi|k^{|\varphi|} + k$, which is exponential in the size of the input and its satisfiability can be checked in polynomial space in the size of the resulting formula [7]. As a result, the problem is in ExpSpace as well.

Finkbeiner and Hahn showed that satisfiability is ExpSpace-complete for sentences of the form $\exists^*\forall^*$ [7]. This implies ExpSpace-hardness of our problem, as if such a sentence, say with k existential quantifiers, is satisfiable then it has a model with at most k traces. □

As the algorithm proceeds by a reduction to the satisfiability problem for $\exists^*$ formulas, which in turn is reduced to LTL satisfiability, one can show that a HyperLTL sentence φ has a model with k traces if and only if it has a model with k ultimately periodic traces.

Next, we consider another variant of the satisfiability problem, where we directly restrict the space of possible models to ultimately periodic ones of the form xy^ω with $|x| + |y| \leq k$. As we encode k in binary, the length of those traces is exponential in the input and the cardinality of the model is bounded doubly-exponentially. This explains the increase in complexity in the following theorem in comparison to Theorem 1.

Theorem 2. *The following problem is* N2ExpTime*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model whose elements are of the form xy^ω with $|x| + |y| \leq k$?*

Proof. For the upper bound, given a HyperLTL sentence φ and $k \in \mathbb{N}$ in binary we start by guessing a model $T \subseteq \{xy^\omega \mid |x| + |y| \leq k\}$. Let $n = \lceil \log_2(k) \rceil + |\varphi|$ be the size of the input. We have $|T| \leq (2^{|\mathrm{AP}|} + 1)^{k+1} \leq (2^n + 1)^{2^n+1}$ (as we can assume all atomic propositions appear in the formula, $|\mathrm{AP}| \leq |\varphi| \leq n$), i.e., $|T|$ is doubly-exponential in n.

Then, we apply to φ a similar transformation as the one used in the proof of Theorem 1, i.e., we create a variable $\pi_{u,v}$ for each trace $uv^\omega \in T$, and we replace in the sentence every universal (existential) quantifier by a conjunction (disjunction) over every possible trace assignment of the quantified variable over T. Formally, we define $\overline{\varphi}$ as follows:

- $\overline{\varphi} = \varphi$ if φ is quantifier-free
- $\overline{\forall \pi.\varphi} = \bigwedge_{uv^\omega \in T} \overline{\varphi}[\pi \leftarrow \pi_{u,v}]$
- $\overline{\exists \pi.\varphi} = \bigvee_{uv^\omega \in T} \overline{\varphi}[\pi \leftarrow \pi_{u,v}]$

The size of the formula is multiplied by $|T|$ at each new quantifier. In the end, the size of $\overline{\varphi}$ is at most $|T|^{|\varphi|}|\varphi| \leq n((2^n+1)^{2^n+1})^n = 2^{2^{O(n)}}$. Define $\Pi_T = [(\pi_{u,v} \to uv^\omega)_{uv^\omega \in T}]$.

Observe that as $T \subseteq \{xy^\omega \mid |x| + |y| \leq k\}$, for all $j > k$, $w(j + k!) = w(j)$ for every $w \in T$. Thus, every quantifier-free formula evaluated over Π_T is satisfied at some index if and only if it is satisfied at some index less than $k! + k$. Therefore, one can evaluate $\overline{\varphi}$ over Π_T by recursively computing for each subformula the set of indices $0 \leq j < k + k!$ such that this subformula is satisfied by $\Pi_T[j, \infty)$. This procedure is polynomial in the size of $\overline{\varphi}$, T and $k!$, all three being doubly-exponential in n, i.e., it is doubly-exponential in the size of the input.

To prove the lower bound, we reduce from the following bounded variant of Post's correspondence problem (PCP): Given two lists $u_1, \ldots, u_n$ and $u'_1, \ldots, u'_n$ of words, does there exist a word $s \in \{1, \ldots, n\}^+$ with $|s| \leq 2^{2^n}$ and $h(s) = h'(s)$? Here h is the homomorphism induced by $m \mapsto u_m$ for every m and h' is defined analogously. We refer to such a word s as a solution. This variant is N2ExpTime-hard, which can be shown by adapting the proof of undecidability of PCP as presented by Hopcroft and Ullman [15]. The same construction allows to reduce from the problem of deciding, given a nondeterministic Turing machine $\mathcal{M}$ and a word w, if $\mathcal{M}$ halts on w in at most $2^{2^{|w|}}$ steps.

Now, given an instance $u_1, \ldots, u_n, u'_1, \ldots, u'_n$ of this problem over an alphabet Σ, we construct a sentence φ and a $k \in \mathbb{N}$ in polynomial time in $\sum_m |u_m| + |u'_m|$ such that φ has a model $T \subseteq \{xy^\omega \mid |x| + |y| \leq k\}$ if and only if the instance has a solution.

Let ℓ be the maximal length of a word of the instance. For all $1 \leq m \leq n$, let $\#(u_m) = u_m \#^{\ell - |u_m|}$ and $\#(u'_m) = u'_m \#^{l - |u'_m|}$, i.e., $|\#(u_m)| = |\#(u'_m)| = \ell$ for all m. For every $a \in \Sigma$, let $\overline{a}$ be a fresh letter and let $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$. For all $1 \leq m \leq n$ and $0 \leq j < |u_m|$ let

$$\#_j(u_m) = \#(u_m)(0) \cdots \#(u_m)(j-1)\overline{\#(u_m)(j)}\#(u_m)(j+1) \cdots \#(u_m)(\ell - 1)$$

be the word $\#(u_m)$ where the letter at position j is replaced by its associated letter in $\overline{\Sigma}$. Analogously, we define $\#_j(u'_m)$.

Further, for all $r \in \mathbb{N}$, let $b_r \in \{0,1\}^*$ be the binary representation of r, with the least significant bit at the beginning and without trailing zeros. Finally, for all $b_r, b_{r'}$ let $[b_r, b_{r'}]$ be the unique word over $\{0,1\}^2$ such that $[b_r, b_{r'}](i) = (b_r(i), b_{r'}(i))$ for all i. If b_r and $b_{r'}$ are not of the same length, then we pad the shorter one with 0's at the end.

In the following, we work with the set $\mathrm{AP} = \Sigma \cup \overline{\Sigma} \cup \{\#, \$, 0, 1\} \cup \{0,1\}^2$ of atomic propositions. In the construction, we ensure that on every trace of the model exactly one proposition is satisfied at any moment in time. Thus, we treat traces as words over AP without making a distinction between a proposition a and the singleton $\{a\}$.

We define two types of traces:

- Traces of type one are of the form $\#(u_m)\#(u'_m)b_r\$^\omega$ for some $1 \leq m \leq n$ and some $r \in \mathbb{N}$, which is called the *rank* of the trace.
- Traces of type two are of the form $(\#_j(u_m))(\#_{j'}(u'_{m'}))[b_r, b_{r'}]\$^\omega$ with $1 \leq m, m' \leq n$, $0 \leq j < |u_m|$, $0 \leq j' < |u'_{m'}|$ and $r, r' \in \mathbb{N}$. Note that we allow $m \neq m'$.

A trace $t_1 = \#(u_{m_1})\#(u'_{m_1})b_{r_1}\$^\omega$ of type one and a trace $t_2 = (\#_j(u_{m_2}))(\#_{j'}(u'_{m'_2}))[b_{r_2}, b_{r'_2}]\$^\omega$ of type two are *u-compatible* if $u_{m_1} = u_{m_2}$ and $r_1 = r_2$ and *u′-compatible* if $u'_{m_1} = u'_{m'_2}$ and $r_1 = r'_2$.

Fix a trace $t_0 = (\#_j(u_m))(\#_{j'}(u'_{m'}))[b_r, b_{r'}]\$^\omega$ of type two. It is *final* if $j = |u_m| - 1$, $j' = |u'_{m'}| - 1$ and $b_r = b_{r'}$. If it is not final, then a trace t_1 of type two is a *successor* of t_0, if one of the following holds:

- $j < |u_m| - 1$ and $j' < |u'_{m'}| - 1$ and $t_1 = (\#_{j+1}(u_m))(\#_{j'+1}(u'_{m'}))[b_r, b_{r'}]\$^\omega$.
- $j = |u_m| - 1$ and $j' < |u'_{m'}| - 1$ and $t_1 = (\#_0(u_{m^*}))(\#_{j'+1}(u'_{m'}))[b_{r+1}, b_{r'}]\$^\omega$ for some $1 \leq m^* \leq n$.
- $j < |u_m| - 1$ and $j' = |u'_{m'}| - 1$ and $t_1 = (\#_{j+1}(u_m))(\#_0(u'_{m^*}))[b_r, b_{r'+1}]\$^\omega$ for some $1 \leq m^* \leq n$.
- $j = |u_m| - 1$ and $j' = |u'_{m'}| - 1$ (which implies $r \neq r'$, as t_0 is not final) and $t_1 = (\#_0(u_{m^*}))(\#_0(u'_{m^{*\prime}}))[b_{r+1}, b_{r'+1}]\$^\omega$ for some $1 \leq m^*, m^{*\prime} \leq n$.

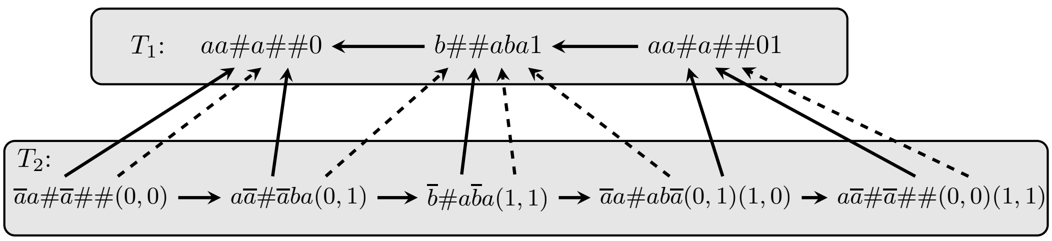
Now, we construct φ. It is the conjunction of sentences expressing the following statements, which can be brought into prenex normal form. In order to improve readability, the construction of the sentences expressing each requirement is left to the reader.

1. Exactly one atomic proposition is satisfied at each position of each trace.
2. Every trace is either of type one or of type two.
3. No two distinct type one traces have the same rank.
4. For every type one trace t of positive rank r there exists a type one trace t' of rank $r - 1$.
5. There exists a type two trace of the form $(\#_0(u_m))(\#_0(u'_m))[b_0, b_0]\$^\omega$ for some m.
6. For every type two trace t_0 there exists a type two trace t_1 such that either t_0 is final or t_1 is a successor of t_0.
7. For every type two trace t_2 there exist two type one traces t_1 and t'_1 such that t_2 and t_1 are u-compatible and t_2 and t'_1 are u'-compatible.
8. In every trace $t = (\#_j(u_m))(\#_{j'}(u'_{m'}))[b_r, b_{r'}]\$^\omega$ of type two, the letters from $\overline{\Sigma}$ in u_m and $u'_{m'}$ are equal.

As an example, consider the following instance of PCP

$$\begin{array}{ll} u_1 = b & u_2 = aa \\ u'_1 = aba & u'_2 = a \end{array}$$

with solution 212. We encode the solution by the set $T_1 \cup T_2$ depicted below (where we drop the suffixes $\$^\omega$ for readability).



Note that T_1 contains type one traces while T_2 contains type two traces. We claim that $T_1 \cup T_2$ satisfies all eight requirement listed above. Requirements 1, 2, 3, and 8 are straightforward to verify. The relation between type one traces specified by Requirement 4 is given by the arrows between the type one traces in T_1. The initial trace as in Requirement 5 is the left most type two trace in T_2 and the successor relation on type two traces as in Requirement 6 is indicated by the arrows between the type two traces. The compatible type one traces as in Requirement 7 are given by the arrows from T_2 to T_1 with the solid arrows denoting the u-compatible type one traces and the dashed arrows denoting the u'-compatible ones.

After this example, we prove that φ has the desired properties in general. To this end, fix $k = 2^n + 2m + 1$, which can be encoded in binary with polynomially many bits in $n + m$.

First, assume there is a solution $s \in \{1, \ldots, n\}^+$, i.e., we have $h(s) = h'(s)$, where h and h' are the homomorphisms induced by mapping m to u_m and u'_m, respectively. We define

$$T_1 = \{\#(u_{s(r)})\#(u'_{s(r)})b_r\$^\omega \mid 0 \leq r < |s|\}$$

and

$$T_2 = \{t_i \mid 0 \leq i < |h(s)| = |h'(s)|\}$$

where

$$t_i = (\#_j(u_{s(r)}))(\#_{j'}(u'_{s(r')}))[b_r, b_{r'}]\$^\omega$$

where r, j are the unique indices such that $r < |s|$, $j < |u_{s(r)}|$, and

$$\sum_{r_0 < r} |h(s(r_0))| + j = i$$

and, similarly, r', j' are the unique indices such that $r' < |s|$, $j' < |u'_{s(r')}|$, and

$$\sum_{r_0 < r'} |h'(s(r_0))| + j' = i.$$

Intuitively, for every i, we pick the pair $(u_{s(r)}, u'_{s(r')})$ of words of the instance the i-th letter of $h(s)$ is in, and mark its position correctly in those words.

Traces of T_1 are of type one and traces of T_2 are of type two. It is then easy to check that Requirements 1 to 8 are satisfied by $T = T_1 \cup T_2$. Furthermore, as $|s| \leq 2^{2^n}$, the size of the b_r is at most 2^n. Thus, all words of T are of the form $x\$^\omega$ with $|x| \leq 2^n + 2m = k - 1$, as required.

Conversely, assume φ is satisfied by a set $T \subseteq \{xy^\omega \mid |x|+|y| \leq k\}$ of traces. Then, by Requirement 2, we have $T = T_1 \cup T_2$ where T_1 is a set of type one traces and T_2 is a set of type two traces. Let r^* be the maximal rank of a trace in T_1 (which is well-defined as T_1 is finite). Then, by Requirements 3 and 4, there exists exactly one trace $t^1_r \in T_1$ such that t^1_r is of rank r, for all $0 \leq r \leq r^*$. Let $s^* \in \{1, ..., n\}^{r^*+1}$ be such that for all $r \leq r^*$, $t^1_r = \#(u_{s^*(r)})\#(u'_{s^*(r)})b_r\$^\omega$. Note that as $t^1_{r^*} = \#(u_{s(r^*)})\#(u'_{s(r^*)})b_{r^*}\$^\omega \in T$, we must have $2m + |b_{r^*}| + 1 \leq k$. Thus, $|b_{r^*}| \leq 2^n$, and $r^* < 2^{2^n}$.

By Requirements 5 and 6, there exists a sequence of traces $t^2_0, t^2_1, \ldots, t^2_p \in T_2$ such that t^2_{i+1} is a successor of t^2_i for all $0 \leq i \leq p$, such that $t^2_0 = (\#_0(u_m))(\#_0(u'_m))[b_0, b_0]\$^\omega$ for some m, and such that t^2_p is final.

An induction over i, using Requirements 6 and 7, shows that the first letter with a bar in t^2_i is the i-th letter of $u_{s^*(0)} \cdots u_{s^*(r^*)}$ and the second one is the i-th letter of $u'_{s^*(0)} \cdots u'_{s^*(r^*)}$. Requirement 8 ensures that those two letters are always equal. As t^2_p is final, there exists a prefix s of s^* such that $|u_{s(0)} \cdots u_{s(|s|-1)}| = |u'_{s(0)} \cdots u'_{s(|s|-1)}| = p+1$ and $u_{s(0)} \cdots u_{s(|s|-1)}(i) = u'_{s(0)} \cdots u'_{s(|s|-1)}(i)$ for all i, i.e., the two words are equal. As $|s| \leq |s^*| \leq 2^{2^n}$, we therefore obtain a solution to the PCP instance. $\square$

As expected, the complexity of the satisfiability problem increases the more traces one has at hand to encode computations. In Theorem 1, we have exponentially many; in Theorem 2, we have doubly-exponentially many. In our last theorem, we consider infinite sets of traces that are finitely representable by finite-state systems. Here, satisfiability becomes intractable, yet still decidable, even when restricted to formulas of temporal depth one.

Formally, a *Kripke structure* $\mathcal{K} = (Q, \delta, q_0, \lambda)$ consists of a finite set Q of states, an initial state $q_0 \in Q$, a transition function $\delta \colon Q \to 2^Q \setminus \{\emptyset\}$ for all q, and a labelling function $\lambda \colon Q \to 2^{\mathrm{AP}}$. A *run* of $\mathcal{K}$ is an infinite sequence $q_0q_1q_2 \cdots$ of states starting with q_0 and such that $q_{j+1} \in \delta(q_j)$ for all $j \in \mathbb{N}$. A trace of $\mathcal{K}$ is the sequence of labels $\lambda(q_0)\lambda(q_1)\lambda(q_2) \cdots$ associated to a run $q_0q_1q_2 \cdots$ of $\mathcal{K}$. The set of traces of $\mathcal{K}$ is denoted by $\mathrm{T}(\mathcal{K})$.

Theorem 3. *The following problem is* TOWER*-complete: Given a HyperLTL sentence φ and $k \in \mathbb{N}$ (in binary), does φ have a model $T(\mathcal{K})$ for some Kripke structure $\mathcal{K}$ with at most k states?*

Proof. Clarkson et al. presented a model-checking algorithm for HyperCTL* (and thus for HyperLTL, which is a fragment of HyperCTL*), and showed that its complexity is a tower of exponentials whose height is the alternation depth of the input sentence [2]. Thus, one can enumerate all Kripke structures with at most k states (up to isomorphism) and model-check them one by one in TOWER. This yields the desired upper bound, as there are "only" exponentially many (in k) Kripke structures with k states.

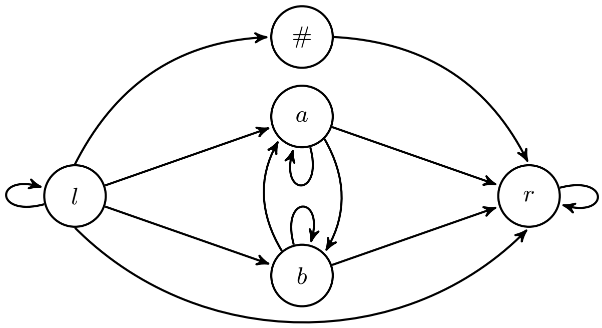


Fig. 1. The Kripke structure $\mathcal{K}$ (all states are initial).

The lower bound is obtained by a reduction from the universality problem for star-free regular expressions with complementation. The equivalence problem for those expressions is TOWER-complete (under elementary reductions, which is standard for TOWER-complete problems), even for two-letter alphabets [22,24]. As those expressions are closed by complementation and union, the universality problem is TOWER-complete as well.

Star-free expressions with complementation over $\{a, b\}$ are generated by the grammar

$$e ::= a \mid b \mid \varepsilon \mid \emptyset \mid e + e \mid ee \mid \neg e$$

and have the obvious semantics inducing a language over $\{a, b\}^*$, denoted by e as well.

Let e be such an expression. We construct a HyperLTL sentence φ_e and a Kripke structure $\mathcal{K}$ such that $T(\mathcal{K})$ is a model of φ_e if and only if e is universal. $\mathcal{K}$ does not depend on e and is shown in Figure 1. As all sets of variables in $\mathcal{K}$ are singletons, we indifferently use the notation a for the letter a and the singleton $\{a\}$. The set of traces induced by this Kripke structure is

$$T(\mathcal{K}) = l^\omega + l^*(a + b)^\omega + l^*(a + b)^* r^\omega + l^*\# r^\omega.$$

Given an expression e and a trace variable π, we inductively define a formula $\psi_{e,\pi}$ which expresses that when π is mapped by a trace assignment Π to a trace of $\mathcal{K}$ of the form $l^n w r^\omega$ with $w \in \{a, b\}^*$, then $w \in e$ if and only if $(T(\mathcal{K}), \Pi) \models \psi_{e,\pi}$.

- $\psi_{\emptyset,\pi} = a_\pi \wedge \neg a_\pi$: No trace assignment satisfies $\psi_{\emptyset,\pi}$, just as the language of $\emptyset$ does not contain any word.
- $\psi_{\varepsilon,\pi} = \mathbf{G}\,(l_\pi \vee r_\pi)$: $(T(\mathcal{K}), \Pi)$ with $\Pi(\pi) = l^n w r^\omega$ satisfies $\psi_{\varepsilon,\pi}$ if and only if $w = \varepsilon$.
- $\psi_{a,\pi} = \exists\tau.(\mathbf{F}\,\#_\tau) \wedge \mathbf{F}\,(a_\pi) \wedge \mathbf{G}\,(l_\tau \Leftrightarrow l_\pi \wedge r_\tau \Leftrightarrow r_\pi)$: The traces of $\mathcal{K}$ with an occurrence of $\#$ are the traces of the form $l^*\# r^\omega$. Thus, $(T(\mathcal{K}), \Pi)$ with $\Pi(\pi) = l^n w r^\omega$ satisfies $\psi_{a,\pi}$ if and only if $l^n w r^\omega$ is a copy of such a trace with $\#$ replaced by a, i.e., if and only if $w = a$.
- $\psi_{b,\pi} = \exists\tau.(\mathbf{F}\,\#_\tau) \wedge \mathbf{F}\,(b_\pi) \wedge \mathbf{G}\,(l_\tau \Leftrightarrow l_\pi \wedge r_\tau \Leftrightarrow r_\pi)$: Similarly to $\psi_{a,\pi}$.
- $\psi_{e_1+e_2,\pi} = \psi_{e_1,\pi} \vee \psi_{e_2,\pi}$.
- $\psi_{e_1 e_2,\pi} = \exists\pi_1.\exists\pi_2.\psi \wedge \psi'$ with

  $$\psi = \mathbf{F}\,r_{\pi_1} \wedge \mathbf{F}\,r_{\pi_2} \wedge \mathbf{G}\,(\neg\#_{\pi_1} \wedge \neg\#_{\pi_2}) \wedge \psi_{e_1,\pi_1} \wedge \psi_{e_2,\pi_2}$$

  expressing that π_1 and π_2 are of the form $l^{n_1} w_1 r^\omega$ and $l^{n_2} w_2 r^\omega$ with $w_1 \in e_1$ and $w_2 \in e_2$, and with

  $$\psi' = \mathbf{G}\,(l_{\pi_2} \Leftrightarrow \neg r_{\pi_1}) \wedge \mathbf{G}\,(a_\pi \Leftrightarrow (a_{\pi_1} \vee a_{\pi_2}) \wedge b_\pi \Leftrightarrow (b_{\pi_1} \vee b_{\pi_2}))$$

  expressing that $n_2 = n_1 + |w_1|$ and that $w = w_1 w_2$, where $\Pi(\pi) = l^n w r^\omega$. Thus, $(T(\mathcal{K}), \Pi)$ satisfies $\psi_{e_1 e_2,\pi}$ if and only if there exist $w_1 \in e_1, w_2 \in e_2$ such that $w = w_1 w_2$.
- $\psi_{\neg e,\pi} = \neg\psi_{e,\pi}$.

Although this inductive definition does not necessarily give a formula in prenex normal form, one can easily check that no quantifier is in the scope of a temporal operator, thus the resulting formula can be turned into a HyperLTL formula.

To conclude, consider the sentence $\varphi_e = \forall\pi.\mathbf{G}\,\neg r_\pi \vee \mathbf{F}\,\#_\pi \vee \psi_{e,\pi}$, which can again be brought into prenex normal form. Further, note that no temporal operator is in the scope of another one, thus φ_e has temporal depth one. The set $T(\mathcal{K})$ is a model of φ_e if and only if all its traces are in $\{a,b,l\}^\omega$, in $l^*\#r^\omega$, or of the form l^*wr^ω with $w \in e$. This is the case if and only if all words $w \in \{a,b\}^*$ are in the language of e, i.e., if and only if e is universal. □

As the Kripke structure $\mathcal{K}$ in the lower bound proof above is fixed, we also obtain a novel hardness result for model-checking.

Corollary 1. *HyperLTL model-checking a fixed Kripke structure with five states is* Tower*-complete, even for sentences of temporal depth one.*

Note that one could already infer the Tower-completeness of the model-checking problem by carefully examining the proof of Theorem 5 of [2] concerning HyperCTL* model-checking. The reduction from the satisfiability problem for QPTL presented there also works for HyperLTL, albeit with temporal depth larger than one. Interestingly, both reductions use a fixed Kripke structure, meaning in particular that the encoding of k has no impact on the asymptotic complexity.

4 Satisfiability for Restricted Classes of Formulas

After studying the HyperLTL satisfiability problem for classes of restricted models, but arbitrary formulas, we now consider restrictions on formulas, but arbitrary models. Recall that Finkbeiner and Hahn presented a complete picture in terms of quantifier prefixes: Satisfiability is PSpace-complete for the alternation-free fragments $\exists^*$ and $\forall^*$ as well as ExpSpace-complete for $\exists^*\forall^*$. In all other cases, the problem is undecidable, i.e., as soon as there is a single universal quantifier in front of existential ones.

In a sense, the decidable fragments are variants of LTL: Both alternation-free fragments can easily be reduced to LTL satisfiability while the $\exists^*\forall^*$ one is easily reducible to the $\exists^*$ fragment, with an exponential blowup. Thus, the decidable fragments barely exceed the realm of LTL.

In this section, we consider another dimension to measure the complexity of formulas, temporal depth, i.e., we restrict the nesting of temporal operators. The hope is that in this setting, we can obtain decidability for larger quantifier prefix classes. However, a slight adaptation of Finkbeiner and Hahn's undecidability result for $\forall\exists^*$, along with an application of Lemma 1 proven below, already shows undecidability for $\forall\exists^*$ formulas of temporal depth two and without untils.

Thus, we have to restrict our search to fragments of temporal depth one, which contain most of the information flow policies expressible in HyperLTL [2]. And indeed, we prove satisfiability decidable for $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F},\mathbf{G})$ formulas. Thus, if the temporal depth is one and untils are excluded, then one can allow a universal quantifier in front of existential ones without losing decidability. This fragment includes, for example, the noninference property [19].

However, even allowing the smallest possible extension, i.e., adding a second universal quantifier, leads again to undecidability: HyperLTL satisfiability is undecidable for $\forall^2\exists^*$ formulas of temporal depth one using only $\mathbf{F}$ as temporal operator. Thus, satisfiability remains hard, even when severely restricting the temporal depth of formulas. Our results for temporal depth one are summarized in Table 1.

Table 1. Complexity of HyperLTL satisfiability in terms of quantifier prefixes and temporal depth. An asterisk * denotes that the upper bound only holds for until-free formulas. All lower bounds in the second column already hold for temporal depth two.

	temporal depth one	arbitrary temporal depth
$\exists^*$ / $\forall^*$	NP-complete ([6]+[7])	PSpace-complete ([7]+[23])
$\exists^*\forall^*$	NExpTime-complete (Thm. 7)	ExpSpace-complete ([7])
$\exists^*\forall\exists^*$	in N2ExpTime* (Thm. 6)	undecidable ([7])
$\forall^2\exists^*$	undecidable (Thm. 5)	undecidable

We begin this section by showing that every HyperLTL formula can be transformed in polynomial time into an equisatisfiable one with quantifier prefix $\forall^2\exists^*$ with temporal depth two. Thus, this fragment already captures the full complexity of the satisfiability problem. This transformation is later used in several proofs.

Theorem 4. *For every HyperLTL sentence one can compute in polynomial time an equisatisfiable sentence of the form* $\forall^2\exists^*$ *with temporal depth at most two.*

We decompose the proof into three steps, formalized by the following three lemmas. We begin by reducing the temporal depth to at most two by adapting a construction of Demri and Schnoebelen, which associates to every LTL formula an equisatisfiable formula with temporal depth at most two [6].

Lemma 1. *For every HyperLTL sentence* $Q_1\pi_1 \dots Q_n\pi_n.\psi$ *with quantifier-free* ψ*, one can compute in polynomial time an equisatisfiable sentence* $Q_1\pi_1 \dots Q_n\pi_n.\exists\pi.\psi'$ *with quantifier-free* ψ' *and temporal depth at most two.*

Proof. Let $\varphi = Q_1\pi_1 \dots Q_n\pi_n.\psi$ with quantifier-free ψ and $Q_i \in \{\exists, \forall\}$ for all i. We denote by $\psi' \sqsubseteq \psi$ the fact that ψ' is a subformula of ψ and introduce a fresh atomic proposition $m^{\psi'}$ for every $\psi' \sqsubseteq \psi$. To every trace assignment $\Pi = [(\pi_i \to t_i)_{1\le i\le n}]$ we associate a *witness trace* $\mathrm{wtns}(t_1, \dots, t_n)$ such that:

- For all $a \in \mathrm{AP}$ and all $j \in \mathbb{N}$, $a \in \mathrm{wtns}(t_1, \dots, t_n)(j)$ if and only if $a \in t_1(j)$, i.e., the projection of $\mathrm{wtns}(t_1, \dots, t_n)$ to AP is equal to t_1.
- For all $\psi' \sqsubseteq \psi$ and all $j \in \mathbb{N}$, $m^{\psi'} \in \mathrm{wtns}(t_1, \dots, t_n)(j)$ if and only if $(T, \Pi[j, \infty)) \models \psi'$.

Thus, $\mathrm{wtns}(t_1, \dots, t_n)$ is a copy of t_1 on which we use the $m^{\psi'}$ to mark the positions at which ψ' is satisfied. Now, define

$$\varphi' = Q_1\pi_1 \dots Q_n\pi_n.\exists\tau.m^{\psi}_{\tau} \wedge \bigwedge_{\psi'\sqsubseteq\psi} \overline{\psi'}.$$

Intuitively, φ' has an additional existentially quantified variable τ that acts as a witness and we ensure that its marking is consistent with the semantics of HyperLTL using a formula $\overline{\psi'}$ for each $\psi' \sqsubseteq \psi$, which is defined as follows:

- $\overline{a_\pi} = \mathbf{G}\,(m^{a_\pi}_{\tau} \Leftrightarrow a_\pi)$
- $\overline{\psi_1 \vee \psi_2} = \mathbf{G}\,(m^{\psi_1\vee\psi_2}_{\tau} \Leftrightarrow (m^{\psi_1}_{\tau} \vee m^{\psi_2}_{\tau}))$
- $\overline{\neg\psi_1} = \mathbf{G}\,(m^{\neg\psi_1}_{\tau} \Leftrightarrow \neg m^{\psi_1}_{\tau})$
- $\overline{\mathbf{X}\,\psi_1} = \mathbf{G}\,(m^{\mathbf{X}\,\psi_1}_{\tau} \Leftrightarrow \mathbf{X}\,m^{\psi_1}_{\tau})$
- $\overline{\psi_1\,\mathbf{U}\,\psi_2} = \mathbf{G}\,(m^{\psi_1\,\mathbf{U}\,\psi_2}_{\tau} \Leftrightarrow m^{\psi_1}_{\tau}\,\mathbf{U}\,m^{\psi_2}_{\tau})$

We claim that φ' has the desired properties.

Suppose φ' is satisfied by a model T'. Then, for all $t' \in T'$ let t be the projection of t' to AP, and let T be the set of those traces t. An induction on ψ allows one to prove that if

$$(T', \Pi') \models \exists\tau.\, m^{\psi}_{\tau} \wedge \bigwedge_{\psi'\sqsubseteq\psi} \overline{\psi'}$$

for some trace assignment Π' over T', then $(T', \Pi') \models \psi$. As ψ only contains propositions from AP, this implies $(T, \Pi) \models \psi$, where for all π, $\Pi(\pi)$ is the projection of $\Pi'(\pi)$ to AP.

Then, an induction on the number of quantifiers allows to generalize this: For all $0 \le i \le n-1$, for all Π', if

$$(T', \Pi') \models Q_{n-i}\pi_{n-i} \dots Q_n\pi_n.\exists\tau.\, m^{\psi}_{\tau} \wedge \bigwedge_{\psi'\sqsubseteq\psi} \overline{\psi'}$$

then $(T, \Pi) \models Q_{n-i}\pi_{n-i} \dots Q_n\pi_n.\psi$ where $\Pi(\pi)$ is again the projection of $\Pi'(\pi)$ to AP. Thus, as T' satisfies φ', T satisfies φ.

Now suppose φ is satisfied by a model T. Let $T' = \{\mathrm{wtns}(t_1, \dots, t_n) \mid t_1, \dots, t_n \in T\}$. As the projection of $\mathrm{wtns}(t_1, \dots, t_n)$ to AP is t_1, T is the set of projections of traces of T' to AP. Furthermore, T' contains a witness for all the tuples of traces of T, allowing us to prove that T' satisfies φ' as follows.

For all Π, if $(T, \Pi) \models \psi$, then let Π' be such that $\Pi'(\pi_i) = \mathrm{wtns}(\Pi(\pi_i), \dots, \Pi(\pi_i))$ (whose projection to AP is $\Pi(\pi_i)$) for all i and $\Pi'(\tau) = \mathrm{wtns}(\Pi(\pi_1), \dots, \Pi(\pi_n))$. An induction on ψ shows that $(T', \Pi') \models m^{\psi}_{\tau} \wedge \bigwedge_{\psi'\sqsubseteq\psi} \overline{\psi'}$. Hence, $(T', \Pi') \models \exists\tau.m^{\psi}_{\tau} \bigwedge_{\psi'\sqsubseteq\psi} \overline{\psi'}$.

Another induction on the number of quantifiers yields that, as T satisfies φ, T' satisfies φ'. Therefore, φ and φ' are equisatisfiable. ◻

Next, we turn the quantifier prefix into the form $\forall^*\exists^*$ without increasing the temporal depth.

Lemma 2. *For every HyperLTL sentence φ, one can compute in polynomial time an equisatisfiable sentence φ' of the form $\forall^*\exists^*$ with $\mathrm{td}(\varphi') = \max(\mathrm{td}(\varphi), 1)$.*

Proof. We call an existential quantifier in a HyperLTL formula *critical* if there is a universal quantifier in its scope. Formulas without critical existential quantifiers are exactly formulas of the form $\forall^*\exists^*$. In what follows, we describe a polynomial time transformation to decrease the number of critical existential quantifiers of a formula. By iterating this procedure, we obtain a formula of the desired form.

Let φ be a HyperLTL formula with at least one critical existential quantifier, i.e., φ is of the form

$$\forall\pi_1 \dots \forall\pi_n.\exists\pi_{n+1} Q_1\tau_1 \dots Q_p\tau_p.\psi$$

with $n \geq 0$, $p > 0$, quantifier-free ψ, $Q_i \in \{\exists, \forall\}$, and $Q_i = \forall$ for some i.

If φ has a model T, then the quantifier prefix $\forall\pi_1 \dots \forall\pi_n.\exists\pi_{n+1}$ of φ induces a Skolem function f associating to every n-tuple $(t_1, \dots, t_n)$ of traces in T a trace $f(t_1, \dots, t_n) = t_{n+1}$ in T such that $(T, \Pi) \models Q_1\tau_1 \dots Q_p\tau_p.\psi$, where Π is the trace assignment with $\Pi(\pi_i) = t_i$ for all $1 \leq i \leq n+1$.

We introduce $n+1$ fresh propositions $m^1, \dots, m^{n+1}$ to mark the traces $t_1, \dots, t_{n+1}$ such that $f(t_1, \dots, t_n) = t_{n+1}$. Let φ'' be the conjunction of the sentences

$$\varphi_1 = \forall\pi_1 \dots \forall\pi_n.\exists\pi_{n+1}.\mathbf{F}\,\Big(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}\Big),$$

which states the existence of the Skolem function, and

$$\varphi_2 = \forall\pi_1 \dots \forall\pi_n.\forall\pi_{n+1}.Q_1\tau_1 \dots Q_p\tau_p.\mathbf{F}\,\Big(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}\Big) \Rightarrow \psi,$$

which ensures that the Skolem function is correct. Note that we replaced the existential quantifier on $\pi n+1$ by a universal one. Note that the conjunction of those sentences can be turned into a HyperLTL sentence by renaming the trace variables π_i in the first one into π'_i and then merging them in the following way:

$$\forall\pi'_1 \dots \forall\pi'_n.\forall\pi_1 \dots \forall\pi_n.\forall\pi_{n+1}.Q_1\tau_1 \dots Q_p\tau_p.\exists\pi'_{n+1}.\mathbf{F}\,\Big(\bigwedge_{i=1}^{n+1} m^i_{\pi'_i}\Big) \wedge \Big[\mathbf{F}\,\Big(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}\Big) \Rightarrow \psi\Big].$$

This formula has one less critical existential quantifier than φ. Hence, it only remains to prove that φ and φ'' are equisatisfiable.

Suppose φ is satisfiable. Then, it has a countable (possibly finite) model T [12]. Thus, for all $t_1, \dots, t_n \in T$, there exists $t_{n+1} \in T$ with $(T, [(\pi_i \to t_i)_{1 \leq i \leq n+1}]) \models Q_1\tau_1, \dots, Q_p\tau_p, \psi$. Thus, for all $t_1, \dots, t_n \in T$ there exists a trace assignment $\Pi_{t_1,\dots,t_n}$ over T with domain $\{\pi_1, \dots, \pi_n, \pi_{n+1}\}$ such that

- $(T, \Pi_{t_1,\dots,t_n}) \models Q_1\tau_1, \dots, Q_p\tau_p, \psi$, and
- $\Pi_{t_1,\dots,t_n}(\pi_i) = t_i$ for all $1 \leq i \leq n$.

Also, fix an injection $g\colon T^{n+1} \to \mathbb{N}$ (which exists as T is countable). Now, let $T'' = \{t'' \mid t \in T\} \subseteq (2^{\mathrm{AP} \cup \{m^i \mid 1 \leq i \leq n+1\}})^\omega$ where t'' is obtained from t as follows:

- The projection of t'' to AP is t, and
- for all $1 \leq i \leq n+1$: $m^i \in t''(j)$ if and only if there exist $t_1, \dots, t_{n+1}$ such that $t = t_i$, $j = g(t_1, ..., t_{n+1})$, and $t_{n+1} = \Pi_{t_1,\dots,t_n}(\pi_{n+1})$, i.e., we encode the Skolem function using the marks m^i at the position given by the injection g to each $(n+1)$-tuple of traces.

We now prove that T'' is a model of φ''. To this end, fix some $t''_1, \dots, t''_{n+1} \in T''$, and let $\Pi'' = [(\pi_i \to t''_i)_{1 \leq i \leq n+1}]$. There exist $t_1, \dots, t_{n+1} \in T$ such that t_i is the projection of t''_i to AP for all i. We denote $\Pi_{t_1,\dots,t_n}$ by Π. If $\Pi(\pi_{n+1}) \neq t_{n+1}$ then $m^i \notin t''_i(g(t_1, ..., t_n, t_{n+1}))$ for some i. By injectivity of g and by definition of T'', there does not exist any other $j \in \mathbb{N}$ such that $m^i \in t''_i(j)$ for all $1 \leq i \leq n+1$. Thus, $(T'', \Pi'') \not\models \mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i})$ i.e., $(T'', \Pi'') \models Q_1\tau_1 \dots Q_p\tau_p.\mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}) \Rightarrow \psi$.

If $\Pi(\pi_{n+1}) = t_{n+1}$, i.e., $\Pi(\pi_i) = t_i$ for all i, then by definition of $\Pi = \Pi_{t_1,\ldots,t_n}$, $(T, \Pi) \models Q_1\tau_1 \ldots Q_p\tau_p.\psi$. As ψ only contains propositions from AP and as t_i is the projection of t''_i to AP for all i, $(T'', \Pi'') \models Q_1\tau_1 \ldots Q_p\tau_p.\psi$. Thus, $(T'', \Pi'') \models Q_1\tau_1 \ldots Q_p\tau_p.\mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}) \Rightarrow \psi$ as well. As we have picked $t_0, \ldots, t_{n+1}$ arbitrarily, we have shown $T'' \models \varphi_2$.

Hence, it remains to consider φ_1. Recall that in this formula, π_{n+1} is quantified existentially. Thus, fix some $t''_1, \ldots, t''_n$ in T'' and let $t_1, \ldots, t_n$ be their projections to AP. As argued above, there is a trace t_{n+1} in T such that $\Pi_{t_1,\ldots,t_n}(\pi_{n+1}) = t_{n+1}$. By definition of g and T'', $m^i \in t''_i(g(t_1, \ldots, t_{n+1}))$ for all i. Therefore, $(T'', [(\pi_i \to t''_i)_{1\le i\le n+1}]) \models \mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i})$. Thus, T'' is also a model of φ_1. Altogether, T'' is a model of φ''.

Conversely, suppose φ'' has a model T''. For all $t'' \in T''$, let t be its projection to AP. Further, let $T = \{t \mid t'' \in T''\}$. Fix some $t_1, \ldots, t_n \in T$. Let $t''_i \in T''$ be such that t_i is the projection of t''_i to AP for all i. As T'' models φ'', there exists $t''_{n+1} \in T''$ such that $(T'', [(\pi_i \to t''_i)_{1\le i\le n+1}]) \models \mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i})$.

Also, as T'' models φ'', $(T'', [(\pi_i \to t'_i)_{1\le i\le n+1}]) \models Q_1\tau_1 \ldots Q_p\tau_p.\mathbf{F}\,(\bigwedge_{i=1}^{n+1} m^i_{\pi_i}) \Rightarrow \psi$. Now, as ψ only contains variables of AP, $(T, [(\pi_i \to t_i)_{1\le i\le n+1}]) \models Q_1\tau_1 \ldots Q_p\tau_p.\psi$.

As we have picked $t_1, \ldots, t_n$ arbitrarily, we have shown that T is a model of φ. □

The construction presented in the proof of Lemma 2 may increase the number of universally quantified variables, but we can decrease that number to two without increasing the temporal or alternation depth. This step also completes the proof of Theorem 4

Lemma 3. *For every HyperLTL sentence φ of the form $\forall^*\exists^*\psi$ with quantifier-free ψ, one can compute in polynomial time an equisatisfiable sentence φ' of the form $\forall^2\exists^*\psi'$ where ψ' is quantifier-free and* $\mathrm{td}(\varphi') = \max(\mathrm{td}(\varphi), 1)$.

Proof. Let $\varphi = \forall\pi_1 \ldots \forall\pi_n \exists\pi_{n+1} \ldots \exists\pi_{n+n'}\psi$ with quantifier-free ψ, i.e., φ starts with n universal quantifiers. We only consider the case $n > 2$, as otherwise we can just choose $\varphi' = \varphi$.

We construct a sentence φ' over AP $\times \{1, \ldots, n\}$, i.e., a single trace represents a tuple of n traces over AP. The two universally quantified variables are employed to ensure that every possible tuple is represented by a trace: Given two traces, we state the existence of n^2 other traces that are shufflings of those two. We also simulate the universal quantifiers of φ using the first of the two universal quantifiers in φ': instead of quantifying n times a single trace, we quantify an n-tuple of traces once.

Formally, we define

$$\varphi' = \forall\pi.\forall\pi'.\exists\pi_{n+1} \ldots \exists\pi_{n+n'}.\exists\tau_{(1,1)}\exists\tau_{(1,2)} \ldots \exists\tau_{(n-1,n)}\exists\tau_{(n,n)}.\psi_1 \wedge \psi_2$$

where

$$\psi_1 = \bigwedge_{i_1=1}^{n} \bigwedge_{i_2=1}^{n} \bigwedge_{a\in \mathrm{AP}} \mathbf{G}\left[\left((a, i_1)_{\tau_{(i_1,i_2)}} \Leftrightarrow (a, i_2)_{\pi'}\right) \wedge \bigwedge_{j\neq i_1} \left((a, j)_{\tau_{(i_1,i_2)}} \Leftrightarrow (a, j)_{\pi}\right)\right]$$

expresses that π takes all the tuples of traces of the model as values: For any two tuples $(t_1, \ldots, t_n)$, $(t'_1, \ldots, t'_n)$ and for all $1 \le i_1, i_2 \le n$ we make sure that the trace $(t_1, \ldots, t_{i_1-1}, t'_{i_2}, t_{i_1+1}, \ldots, t_n)$ also appears. This property implies that all tuples of traces are taken into account when π is ranging over them.

Furthermore, ψ_2 is obtained from ψ by replacing each atomic formula a_{π_j} for $1 \le j \le n$ with $(a, j)_\pi$ and every a_{π_j} for $n+1 \le j \le n+n'$ by $(a, 1)_{\pi_j}$, i.e., we identify the universally quantified traces with the components of a tuple of traces assigned to π in ψ_2 and the existentially ones with the first components of the tuples assigned to the π_i in ψ_2.

Suppose φ has a model T. Then, for all $t_1, \ldots, t_n \in T$ let $\mathrm{mrg}(t_1, \ldots, t_n)$ be the trace over AP $\times \{1, \ldots, n\}$ such that $\mathrm{mrg}(t_1, \ldots, t_n)(j) = \bigcup_{i=1}^{n} t_i(j) \times \{i\}$ for all $j \in \mathbb{N}$. Let $\mathrm{mrg}(T) = \{\mathrm{mrg}(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in T\}$. Note that there is a bijection between n-tuples over traces from T and traces in $\mathrm{mrg}(T)$.

Let $t' \in \mathrm{mrg}(T)$, there exist $t_1, \ldots, t_n \in T$ such that $t' = \mathrm{mrg}(t_1, \ldots, t_n)$. As T is a model of φ, there exist $t_{n+1}, \ldots, t_{n+n'}$ such that $(T, [(\pi_i \to t_i)_{1\le i\le n+n'}]) \models \psi$. Further, let $t'_1, \ldots, t'_n \in T$ be arbitrary. Then, one can check that the trace assignment associating

- $\mathrm{mrg}(t_1, \ldots, t_n)$ to π,
- $\mathrm{mrg}(t'_1, \ldots, t'_n)$ to π',

– $\mathrm{mrg}(t_i, \ldots, t_i)$ to π_i with $n+1 \leq i \leq n+n'$, and
– $\mathrm{mrg}(t_1, \ldots, t_{i_1-1}, t'_{i_2}, t_{i_1+1}, \ldots, t_n)$ to $\tau_{(i_1,i_2)}$

satisfies both ψ_1 and ψ_2. Thus, as we have picked the t_i and t'_i for $1 \leq i \leq n$ arbitrarily, we conclude that $\mathrm{mrg}(T)$ is a model of φ'.

Conversely, suppose φ' has a model T'. For all $t' \in T'$, let t be the trace over AP such that $t(j) = \{a \mid (a,1) \in t'(j)\}$ for all $j \in \mathbb{N}$, and let $T = \{t \mid t' \in T'\}$. As T' satisfies φ', for all $t'_1, t'_2 \in T$, for all $1 \leq i_1, i_2 \leq n$ there exist t'_{i_1,i_2} which is identical to t'_1 on all $\mathrm{AP} \times \{i\}$ except for $\mathrm{AP} \times \{i_1\}$, for which it is identical to t'_2 on $\mathrm{AP} \times \{i_2\}$. Because of this property, one can see that T' is equal to $\{\mathrm{mrg}(t_1, \ldots, t_n) \mid t_1, \ldots, t_n \in T\}$.

Now, an induction on the construction of ψ_2 shows that, as T' satisfies φ', T satisfies φ. □

Thus, $\forall^2\exists^*$ formulas with temporal depth two capture the complete complexity of the satisfiability problem for HyperLTL. As the latter problem is undecidable and as all reduction presented above are effective, we immediately obtain that satisfiability for $\forall^2\exists^*$ formulas with temporal depth two is also undecidable.

As alluded to above, an even stronger result can be obtained by strengthening the proof of Hahn and Finkbeiner for $\forall\exists^*$ formulas to only use temporal depth two. Thus, only formulas of temporal depth one remain to consider.

Before we start investigating this class let us quickly comment on why we disregard temporal depth zero: every such sentence can easily be turned to an equisatisfiable instance of QBF, which is known to be solvable in polynomial space.

Thus, it only remains to consider formulas with arbitrary quantifier prefixes, but temporal depth one. Our main result of this section shows that even this problem is undecidable, even for $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$ formulas with alternation depth one. Due to the restriction on the temporal depth, our encoding of a Minsky machine is more complicated than it would be with arbitrary temporal depth.

Theorem 5. *The following problem is undecidable: Given a $\forall^2\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G})$ sentence φ, is φ satisfiable?*

Proof. We reduce from the (non)-halting problem for 2-counter Minsky machines. Recall that such a machine can be seen as a tuple $\mathcal{M} = (Q, \Delta, q_0)$ where Q is a finite set of states, $q_0 \in Q$ an initial state, and $\Delta \subseteq Q \times \{1,2\} \times OP \times Q$ a set of transition rules, where $OP = \{\texttt{++}, \texttt{--}, \texttt{=0?}\}$. A configuration of $\mathcal{M}$ is an element of $Q \times \mathbb{N} \times \mathbb{N}$. For all $n, n' \in \mathbb{N}$, $op \in OP$ we write $n \xrightarrow{op} n'$ if:

– op is $\texttt{++}$ and $n' = n + 1$.
– op is $\texttt{--}$ and $n' = n - 1$ (note that this operation is only applicable if $n > 0$).
– op is $\texttt{=0?}$ and $n' = n = 0$.

There is a transition from (q, n_1, n_2) to (q', n'_1, n'_2) if and only if there exists $i \in \{1,2\}$ and $op \in OP$ such that $(q, i, op, q') \in \Delta$, $n_{3-i} = n'_{3-i}$ and $n_i \xrightarrow{op} n'_i$. Given such a machine, it is undecidable whether it has an infinite computation $(q_0, 0, 0) \to (q_1, n_1^1, n_2^1) \to (q_2, n_1^2, n_2^2) \to \cdots$.

Let $\mathcal{M} = (Q, \Delta, q_0)$ be a 2-counter Minsky machine. We use $\mathrm{AP} = Q \cup \{1,2\}$ as atomic propositions. Given $i \in \{1,2\}$, we denote by $\bar{i}$ the other proposition. Consider the formula $\psi_1 = \forall\pi.\forall\pi'.\mathbf{G}\,(1_\pi \Rightarrow 1_{\pi'}) \vee \mathbf{G}\,(1_{\pi'} \Rightarrow 1_\pi)$. We define ψ_2 with $2 \in \mathrm{AP}$ analogously. In the following, we only consider sets of traces that satisfy $\psi_1 \wedge \psi_2$.

For each trace $t \in (2^{\mathrm{AP}})^\omega$ and $i \in \{1,2\}$, we define the *i-set* of t as $S_i(t) = \{j \in \mathbb{N} \mid i \in t(j)\}$. Now fix $T \subseteq (2^{\mathrm{AP}})^\omega$ that satisfies $\psi_1 \wedge \psi_2$. We define the order $\leq_i$ on T as follows: for all $t, t' \in T$, $t \leq_i t'$ if and only if $S_i(t) \subseteq S_i(t')$. We write $t <_i t'$ if $S_i(t) \subsetneq S_i(t')$. As T satisfies $\psi_1 \wedge \psi_2$, the $\leq_i$ are total orders on T. We also define for all $t \in T$ and $i \in \{1,2\}$, the *rank* of t with respect to i as $\mathrm{rk}_i(t) = |\{S_i(t') \mid t' \in T \text{ and } t' <_i t\}|$, which may be infinite. Note that if $S_i(t) = \emptyset$ then $\mathrm{rk}_i(t) = 0$, and that if $S_i(t) = S_i(t')$ then $\mathrm{rk}_i(t) = \mathrm{rk}_i(t')$. Finally, as $\leq_i$ is a total order, if we have $t <_i t'$, but there is no t'' with $t <_i t'' <_i t'$, then $\mathrm{rk}_i(t') = \mathrm{rk}_i(t) + 1$. Note that this holds even when $\mathrm{rk}_i(t)$ is infinite, assuming $\infty + 1 = \infty$.

We construct a $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$ formula φ that encodes the existence of an infinite computation $(q_0, 0, 0) \to (q_1, n_1^1, n_2^1) \to (q_2, n_1^2, n_2^2) \to \cdots$ of $\mathcal{M}$. In a model T of φ, a configuration (q, n_1, n_2) is encoded by a trace t with $t(0) \cap Q = \{q\}$ and for $i \in \{1,2\}$, $rk_i(t) = n_i$. Then, φ states the existence of an initial trace t_0, representing the configuration $(q_0, 0, 0)$, as well as the existence of a successor t'

encoding (q', n_1', n_2') for each trace t encoding (q, n_1, n_2), i.e., we require $(q, n_1, n_2) \to (q', n_1', n_2')$. The latter is witnessed by the existence of a transition (q, i, op, q') such that:

1. $t(0) \cap Q = \{q\}$ and $t'(0) \cap Q = \{q'\}$, i.e., t and t' indeed encode the states of their respective configurations correctly.
2. For all $j \in \mathbb{N}$, $\overline{i} \in t(j)$ if and only if $\overline{i} \in t'(j)$, i.e. $S_{\overline{i}}(t) = S_{\overline{i}}(t')$. Thus, as argued above, $\mathrm{rk}_{\overline{i}}(t) = \mathrm{rk}_{\overline{i}}(t')$, which implies $n_{\overline{i}} = n'_{\overline{i}}$.
3. If op is `++` then $t <_i t'$ and there does not exist any t'' such that $t <_i t'' <_i t'$, i.e., $\mathrm{rk}_i(t') = \mathrm{rk}_i(t) + 1$, as $\leq_i$ is a total order. Then, we have $n_i' = n_i + 1$.
4. If op is `--` then $t >_i t'$ and there does not exist any t'' such that $t >_i t'' >_i t'$, i.e., $\mathrm{rk}_i(t') = \mathrm{rk}_i(t) - 1$, as $\leq_i$ is a total order. Then, we have $n_i' = n_i - 1$.
5. If op is `=0?` then for all $j \in \mathbb{N}$, $i \notin t(j)$ and $i \notin t'(j)$. Hence, $S_i(t) = S_i(t') = \emptyset$, i.e., $\mathrm{rk}_i(t) = \mathrm{rk}_i(t') = 0$, which implies $n_i = n_i' = 0$.

We encode those conditions in φ, which is the conjunction of the following three sentences and of $\psi_1 \wedge \psi_2$:

- $\varphi_1 = \forall \pi. \bigwedge_{q \neq q' \in Q} q_\pi \Rightarrow \neg q'_\pi$ expresses that a trace is associated to at most one state.
- $\varphi_2 = \exists \pi_0. (q_0)_{\pi_0} \wedge \mathbf{G}\, (\neg 1_{\pi_0} \wedge \neg 2_{\pi_0})$ expresses the existence of a trace representing the initial configuration $(q_0, 0, 0)$.
- $\varphi_3 = \forall \pi. \exists \pi'. \bigvee_{(q,i,op,q') \in \Delta} q_\pi \wedge q'_{\pi'} \wedge \varphi_{i,op} \wedge \mathbf{G}\, (\overline{i}_\pi \Leftrightarrow \overline{i}_{\pi'})$ expresses that all traces have a successor obtained by faithfully simulating a transition of the machine.

Here, we use the formulas

- $\varphi_{1,\texttt{++}} = \forall \pi''. \pi <_1 \pi' \wedge (\pi'' \leq_1 \pi \vee \pi' \leq_1 \pi'')$,
- $\varphi_{1,\texttt{--}} = \forall \pi''. \pi >_1 \pi' \wedge (\pi'' \geq_1 \pi \vee \pi' \geq_1 \pi'')$, and
- $\varphi_{1,\texttt{=0?}} = \mathbf{G}\, (\neg 1_\pi \wedge \neg 1_{\pi'})$,

where $\pi \leq_1 \pi' = \mathbf{G}\, (1_\pi \Rightarrow 1_{\pi'})$ and $\pi <_1 \pi' = \pi \leq_1 \pi' \wedge \mathbf{F}\, (\neg 1_\pi \wedge 1_{\pi'})$. Finally, we define the formulas $\leq_2$, $<_2$, and $\varphi_{2,op}$ analogously.

The sentence φ is not in prenex normal form. However, as no quantifier appears in the scope of a temporal operator, it can be put in that form. Further, it is not of the form $\forall^2\exists^*$, but we can apply Lemmas 2 and 3 to bring it into this form while preserving the temporal depth, which is already one. We claim that φ is satisfiable if and only if $\mathcal{M}$ has an infinite computation starting in $(q_0, 0, 0)$.

Suppose φ is satisfied by a model T. The subformulas φ_1 and φ_2 enforce that T contains a trace t_0 encoding the initial configuration $(q_0, 0, 0)$ of $\mathcal{M}$. Further, φ_3 expresses that every trace t encoding a configuration (q, n_1, n_2) has a successor t' encoding a configuration (q', n_1', n_2') with $(q, n_1, n_2) \to (q', n_1', n_2')$. Thus, there exists an infinite sequence $t_0, t_1, t_2, \ldots$ of traces encoding an infinite run of $\mathcal{M}$.

Conversely, suppose $\mathcal{M}$ has an infinite run $(q_0, 0, 0) \to (q_1, n_1^1, n_2^1) \to (q_2, n_1^2, n_2^2) \cdots$, then for all j let t_j be the trace whose projection to Q is $\{q_j\}\emptyset^\omega$, and whose projection to $\{i\}$ is $\{i\}^{n_i^j}\emptyset^\omega$ for $i \in \{1, 2\}$. One can then easily check that $\{t_j \mid j \in \mathbb{N}\}$ is a model of φ. $\square$

Thus, two universal quantifiers before some existential ones and using only $\mathbf{F}$ and $\mathbf{G}$ without nesting yields undecidable satisfiability. Our next result shows that removing one of the two universal quantifiers allows us to recover decidability, even when allowing nested next operators and leading existential quantifiers.

As a first step in the proof, we show that the nested next operators can be eliminated without introducing additional universal quantifiers. This is true, as we are only interested in satisfiability.

Lemma 4. *For every $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ sentence, one can construct in polynomial time an equisatisfiable $\exists^*\forall\exists^*$ HyperLTL$^1(\mathbf{F}, \mathbf{G})$ sentence.*

Proof. Let $\varphi = \exists \tau_1 \ldots \exists \tau_n. \forall \pi. \exists \tau_{n+1} \ldots \exists \tau_{n+n'}. \psi$ be a HyperLTL$^1(\mathbf{F}, \mathbf{G}, \mathbf{X}^*)$ sentence. We assume w.l.o.g., that ψ is a Boolean combination of formulas of the form $\mathbf{F}\, \beta$ and $\mathbf{X}^k \beta$ where β is a Boolean combination of atomic propositions. Let d be the maximal integer such that ψ contains a subformula of the form $\mathbf{X}^d \beta$ (which is 0, if there is no $\mathbf{X}$ in ψ).

We extend the set of atomic propositions with fresh ones $m^0, \ldots, m^d$ and define

$$\varphi' = \exists \tau_0. \exists \tau_1 \ldots \exists \tau_n. \forall \pi. \exists \tau_{n+1} \ldots \exists \tau_{n+n'}. \psi'$$

(i.e., we just add one existentially quantified variable τ_0) with

$$\psi' = \bigwedge_{k=0}^{d} \left(\mathbf{F}\, m^k_{\tau_0}\right) \wedge \left(\mathbf{F}\,(\bigvee_{a \in \mathrm{AP} \cup \{m^0, \ldots, m^d\}} a_\pi \oplus a_{\tau_0}) \Rightarrow \psi''\right)$$

where ψ'' is ψ where every maximal subformula of the form $\mathbf{X}^k\beta$ has been replaced with $\mathbf{G}\,({m^k}_{\tau_0} \Rightarrow \beta)$. Intuitively, we state the existence of a special trace on which the atomic propositions $m^0, \ldots, m^d$ mark some positions which will simulate the positions from 0 up to d in any order with any multiplicity. As we are dealing with satisfiability, and as the order of the valuations at these positions is irrelevant for the satisfaction of a HyperLTL^1($\mathbf{F}, \mathbf{G}$) formula, we obtain an equisatisfiable formula.

If φ has a model T, then let $T' = T \cup \{t_0\}$, with $t_0 = \{m^0\} \ldots \{m^d\}\emptyset^\omega$. Here every m^k appears once, at position k, thus, $\mathbf{X}^k\beta$ is satisfied by some assignment over T if and only $\mathbf{G}\,(m^k \Rightarrow \beta)$ is satisfied by the same assignment over T'. Then, one can show by a simple induction over the construction of ψ' that for all $t_1, \ldots, t_{n+n'}, t$, $(T, [(\tau_i \to t_i)_{1 \le i \le n+n'}, \pi \to t])$ satisfies ψ if and only if $(T', [(\tau_i \to t_i)_{0 \le i \le n+n'}, \pi \to t])$ satisfies ψ'. Since φ and φ' have the same quantifier prefix (but for the quantification of τ_0), and as T satisfies φ, T' satisfies φ'.

Conversely, if φ' has a model T', then let $t'_0 \in T'$ be such that $(T', [\tau_0 \to t'_0])$ satisfies

$$\exists \tau_1 \ldots \exists \tau_m . \forall \pi . \exists \tau_{n+1} \ldots \exists \tau_{n+n'} . \psi'.$$

In particular $(T', [\tau_0 \to t'_0])$ satisfies $\bigwedge_{k=0}^{d} \mathbf{F}\, m^k_{\tau_0}$. Thus, there exist $j_0, \ldots, j_k \in \mathbb{N}$ such that $m^k \in t'_0(j_k)$ for all k. For all $t' \in T' \setminus \{t'_0\}$ let t be the projection to AP of $t'(j_0) \cdots t'(j_d)t'$, and let $T = \{t \mid t' \in T' \setminus \{t'_0\}\}$. Thus, if a trace assignment over T' satisfies $\mathbf{G}\,(m^k_{\tau_0} \Rightarrow \beta)$ then the associated trace assignment over T satisfies $\mathbf{X}^k\beta$. Furthermore, if a trace assignment over T' satisfies $\mathbf{F}\,\beta$ (respectively $\mathbf{G}\,\beta$) then so does the associated trace assignment over T (we have not modified the set of combinations of valuations that eventually appear, just their order). Hence, as ψ can be seen as a positive Boolean combination of such formulas, an induction over φ shows that, as φ' is satisfiable, so is φ. ◻

Now, we are ready to prove our main decidability result in this section. Note that we do not claim a matching lower bound here. We comment on this gap in the conclusion.

Theorem 6. *The following problem is in* N2ExpTime*: Given a HyperLTL*^{1}($\mathbf{F}, \mathbf{G}, \mathbf{X}^*$) *sentence* φ *of the form* $\exists^*\forall\exists^*$*, is* φ *satisfiable?*

Proof. Let $\varphi = \exists \tau_1 \ldots \tau_n . \forall \pi . \exists \tau_{n+1} \ldots \exists \tau_{n+n'} . \psi$ be a HyperLTL^1($\mathbf{F}, \mathbf{G}, \mathbf{X}^*$) sentence with quantifier-free ψ. Due to Lemma 4, it is enough to consider the case where ψ is a Boolean combination of formulas of the form $\mathbf{F}\,\beta$ for a Boolean combination β of atomic propositions.

To every tuple $(t_1, \ldots, t_k)$ of traces $t_i \in (2^{\mathrm{AP}})^\omega$, we associate a finite set of tuples of valuations $V(t_1, \ldots, t_k) = \{(t_1(j), \ldots, t_k(j)) \mid j \in \mathbb{N}\} \subseteq (2^{\mathrm{AP}})^k$, i.e., the set all the tuples of valuations that appear eventually. The cardinality of $V(t_1, \ldots, t_k)$ is at most $2^{k|\mathrm{AP}|}$.

Let β be a Boolean combination of atomic propositions over trace variables $\pi_1, \ldots, \pi_k$. Then, a trace assignment $[(\pi_i \to t_i)_{1 \le i \le k}]$ satisfies $\mathbf{F}\,\beta$ if and only if there exists $j \in \mathbb{N}$ such that β is satisfied at position j of $(t_1, \ldots, t_k)$, i.e., there exists $(v_1, \ldots, v_k) \in V(t_1, \ldots, t_k)$ such that $(v_1, \ldots, v_k)$ satisfies β (in the sense that any trace assignment Π such that $\Pi(\pi_i)(0) = v_i$ for all i satisfies β). Intuitively, we abstract a tuple of traces into a finite set of tuples of valuations, and then abstract a model as a set of such finite representations. Then, we show that satisfiability can be decided using such abstractions.

As a consequence, whether a given trace assignment $[(\pi_i \to t_i)_{1 \le i \le k}]$ satisfies a given Boolean combination ψ of formulas $\mathbf{F}\,\beta$ only depends on $V(t_1, \ldots, t_k)$, and given $V \subseteq (2^{\mathrm{AP}})^k$, one can check in polynomial time whether a trace assignment yielding V satisfies ψ. If it is the case, we say that V satisfies ψ.

To check the satisfiability of φ, we start by nondeterministically guessing a set $S \subseteq 2^{(2^{\mathrm{AP}})^{n+n'+1}}$ of sets of $(n + n' + 1)$-tuples of valuations. This set is supposed to represent a model of φ. The n first valuations represent the fixed values assigned to $\tau_1, \ldots, \tau_n$. The $(n + 1)$-th represents the valuation of the universally quantified variable. Thus, for every trace of the model there must exist a tuple in which that trace is represented at position $n + 1$. The valuations of positions $n + 2$ to $n + n'$ have to be such that φ is satisfied by all tuples.

Thus, we check the following requirements:

1. For all $V_1, V_2 \in S$, $\{(v_1, \dots, v_n) \mid (v_1, \dots, v_{n+n'+1}) \in V_1\}$ is equal to $\{(v_1, \dots, v_n) \mid (v_1, \dots, v_{n+n'+1}) \in V_2\}$: The set of values taken by the traces assigned to $\tau_1, \dots, \tau_n$ cannot depend on the values of the other variables. Thus, we ensure that these values are fixed in the guessed model.
2. For all $V \in S$ and $1 \leq i \leq n+n'+1$, there exists $V' \in S$ such that $\{(v_1, \dots, v_n, v_i) \mid (v_1, \dots, v_{n+n'+1}) \in V\} = \{(v_1, \dots, v_{n+1}) \mid (v_1, \dots, v_{n+n'+1}) \in V'\}$. All the values taken by the existentially quantified variables have to be taken by the universally quantified one as well.
3. For all $V \in S$, V satisfies ψ.

If all requirements are satisfied, we accept, otherwise we reject. This procedure requires nondeterministic doubly-exponential time as $|S| \leq 2^{2^{|\mathrm{AP}|+n+n'+1}}$.

Suppose φ is satisfiable and fix a model T. There exist $t_1, \dots, t_n \in T$ such that $(T, [(\tau_i \to t_i)_{1 \leq i \leq n}]) \models \forall \pi \exists \tau_{n+1} \dots \exists \tau_{n+n'}.\psi$. Furthermore, for a fixed $t \in T$ there exist $t_{n+1}, \dots, t_{n+n'} \in T$ such that $(T, [(\tau_i \to t_i)_{1 \leq i \leq n+n'}, \pi \to t]) \models \psi$. Let $V^*(t) = \{(t_1(j), \dots, t_n(j), t(j), t_{n+1}(j), \dots, t_{n+n'}(j)) \mid j \in \mathbb{N}\}$.

Now, one can easily check that Requirements 1, 2, and 3 are satisfied by $\{V^*(t) \mid t \in T\}$. Thus, the algorithm accepts φ.

Conversely, suppose the algorithm accepts φ. Then, there exists some S satisfying all three requirements above. We construct from S a model T of φ.

Let $t_1, \dots, t_n$ be traces such that for all $V \in S$, $\{(v_1, \dots, v_n) \mid (v_1, \dots, v_{n+n'+1}) \in V\} = V(t_1, \dots, t_n)$, and for all $(v_1, \dots, v_{n+n'+1}) \in V$, $(v_1, \dots, v_n) = (t_1(j), \dots, t_n(j))$ for infinitely many j, i.e., each of the valuations appears infinitely often in the traces. Those traces can be constructed due to Requirement 1.

Let $T_0 = \{t_1, \dots, t_n\}$. For all $\ell \in \mathbb{N}$ we construct T_ℓ by induction on $\ell \in \mathbb{N}$, while maintaining the following two invariants:

1. For all $t \in T_\ell$ there exists $V \in S$ such that $V(t_1, \dots, t_n, t)$ is equal to $\{(v_1, \dots, v_{n+1}) \mid (v_1, \dots, v_{n+n'+1}) \in V\}$, and for all $(v_1, \dots, v_{n+n'+1}) \in V$, $(v_1, \dots, v_{n+1})$ is equal to $(t_1(j), \dots, t_n(j), t(j))$ for infinitely many j, where the t_i are the traces in T_0.
2. If $\ell > 0$ then for every $t \in T_{\ell-1}$, there exist traces $t_{n+1}, \dots, t_{n+n'} \in T_\ell$ such that $[(\tau_i \to t_i)_{1 \leq i \leq n+n'}, \pi \to t] \models \psi$.

By Requirement 2 and by construction, T_0 satisfies Invariant 1, and it clearly satisfies Invariant 2. Let $\ell \in \mathbb{N}$, suppose T_ℓ has been constructed, and that it satisfies Invariants 1 and 2. By Invariant 1, for all $t \in T_\ell$ we can construct traces $t_{n+1}, \dots, t_{n+n'}$ such that $V(t_1, \dots, t_n, t, t_{n+1}, \dots, t_{n+n'}) \in S$ and for all $(v_1, \dots, v_n, v, v_{n+1}, \dots, v_{n+n'}) \in V(t_1, \dots, t_n, t, t_{n+1}, \dots, t_{n+n'})$, it is the case that $(v_1, \dots, v_n, v, v_{n+1}, \dots, v_{n+n'})$ is equal to
$(t_1(j), \dots, t_n(j), t(j), t_{n+1}(j), \dots, t_{n+n'}(j))$ for infinitely many j (as all the $(v_1, \dots, v_n, v)$ appear infinitely many times in $(t_1, \dots, t_n, t)$ by Invariant 1). Let $I(t) = \{t_{n+1}, \dots, t_{n+n'}\}$. Let $T_{\ell+1} = \bigcup_{t \in T} I(t)$, which satisfies Invariant 1 by Requirement 2. It also satisfies Invariant 2 by definition. Furthermore, by Requirement 3, $V(t_1, \dots, t_n, t, t_{n+1}, \dots, t_{n+n'})$ satisfies ψ.

Finally, let $T = \bigcup_{\ell \in \mathbb{N}} T_\ell$ and let $t \in T$. Then, there exists an ℓ such that $t \in T_\ell$. Thus, there also exist $t_{n+1}, \dots, t_{n+n'} \in T_{\ell+1}$ such that $[(\tau_i \to t_i)_{1 \leq i \leq n+n'}, \pi \to t]$ satisfies ψ. Therefore, T satisfies φ. $\square$

Recall that satisfiability of $\exists^*\forall^*$ formulas is ExpSpace-complete [7]. The proof of Finkbeiner and Hahn can be slightly adapted to produce a formula of temporal depth two: their approach states the existence of a trace representing a sequence of configurations of an exponential-space bounded Turing machine. The only difficulty that can arise in expressing the correctness of the run described by that trace is relating a position of one of the configurations to the neighbouring positions in the next configuration (in order to simulate the movement of the head). One may then require to combine an until and a next in order to express this requirement, in the scope of an always expressing that it holds for every position. This nesting can be removed by adding a fresh proposition p that is satisfied on all positions of the first configuration, on none of the second one, and so on, i.e., its truth value alternates between the configurations. One can then express the previous requirement with a single until in the scope of an always, yielding temporal depth two.

Our next result shows that one obtains slightly better complexity when restricting the temporal depth of formulas to one.

Theorem 7. *The following problem is* NExpTime*-complete: Given an $\exists^*\forall^*$ HyperLTL sentence φ with temporal depth one, is φ satisfiable?*

Proof. This proof is an adaptation of the proof by Finkbeiner and Hahn showing that the problem is ExpSpace-complete for sentences of arbitrary temporal depth [7]. For the upper bound, we use the transformation of Finkbeiner and Hahn to turn the initial sentence into an equisatisfiable exponentially larger one of the form $\exists^*\psi'$ without increasing the temporal depth, by replacing universal quantifiers with conjunctions. Then, we turn that sentence into an equisatisfiable LTL formula of the same size and the same temporal depth by merging traces, again due to Finkbeiner and Hahn. Finally, we apply a result of Demri and Schnoebelen stating that satisfiability of LTL formulas of temporal depth one is in NP [6]. Altogether, we obtain the desired algorithm.

For the lower bound, we again adapt ideas of Finkbeiner and Hahn. We reduce from the following NExpTime-complete problem: Given a nondeterministic Turing machine $\mathcal{M}$ and an integer n (in unary), does $\mathcal{M}$ accept the empty word in time at most 2^n? Given such a machine $\mathcal{M} = (Q, \Sigma, \Delta, q_0, q_f)$, we construct a HyperLTL$^1(\mathbf{F},\mathbf{G})$ formula φ of the form $\exists^*\forall^*\psi$ such that φ is satisfiable if and only if $\mathcal{M}$ accepts the empty word in less than 2^n steps. We can assume that there is a loop on the final state, i.e., the machine accepts in at most 2^n steps if and only if it accepts in exactly 2^n steps. We also assume Σ to contain a blank letter B meaning that nothing has been written at that position yet. In particular, the initial tape contains only B.

In our encoding of a run of $\mathcal{M}$, the valuation at some position encodes the content of one cell at one moment in time. Thus, we use the set $Q \cup \Sigma \cup \{\mathrm{spc}_1, \ldots, \mathrm{spc}_n\} \cup \{\mathrm{tm}_1, \ldots, \mathrm{tm}_n\} \cup \{h\} \cup \{0, 1\}$ of atomic propositions, where h, spc_i, and tm_i are fresh atomic propositions representing respectively whether the head is at the encoded cell, the position on the tape and time (the two latter in binary). For all $0 \le j < 2^n$ let $\mathit{time}(j)$ and $\mathit{space}(j)$ be the valuations over the tm_i and spc_i representing j.

To encode the existence of an accepting run of $\mathcal{M}$, we require the existence of a trace t encoding such a run in the following sense:

- Every valuation of $\mathrm{tm}_1, \ldots, \mathrm{tm}_k, \mathrm{spc}_1, \ldots, \mathrm{spc}_k$ appears eventually on t.
- Two positions with the same valuation of the tm_i and spc_i also have the same valuation of the other atomic propositions.
- For all $j \in \mathbb{N}$, $t(j)$ contains exactly one letter from Σ and one state from Q.
- There exists a run of $\mathcal{M}$ such that for all j_t, j_s, if at the j_t-th step the machine is in state q and has an a at position j_s, then for all $m \in \mathbb{N}$, if $t(m) \cap \{\mathrm{tm}_i \mid 1 \le i \le n\} = \mathit{time}(j_t)$ and $t(m) \cap \{\mathrm{spc}_i \mid 1 \le i \le n\} = \mathit{space}(j_s)$ then $a, q \in t(m)$ and $h \in t(m)$ if and only if the head is at position j_s at the j_t-th step of the run, i.e., every trace faithfully represents the content of the cell at the step of the run and the position on the tape it is assigned to.

Now, we define φ as

$$\exists\pi.\exists\pi_0.\exists\pi_1.\forall\tau_1\ldots\forall\tau_n.\forall\tau'_1\ldots\forall\tau'_n.\forall\sigma_1\ldots\forall\sigma_n.\forall\sigma'_1\ldots\sigma'_n.$$
$$\psi_{0/1} \wedge [\psi_{assign} \Rightarrow (\psi_{halt} \wedge \psi_{init} \wedge \psi_{cons} \wedge \psi_{complete} \wedge \bigvee_{(q,q',a,d)\in\Delta} \psi_{trans}(q, q', a, a', d))].$$

We explain the role of φ's subformulas below, but leave the construction of the formulas to the reader in order to improve readability. We just remark that all can be constructed with temporal depth one.

- $\psi_{0/1}$ expresses that π_0 takes value $\{0\}^\omega$ and that π_1 takes value $\{1\}^\omega$.
- ψ_{assign} ensures that all the traces assigned to the $\tau_i, \tau'_i, \sigma_i, \sigma'_i$ are equal to either the trace assigned to π_0 or the one assigned to π_1. The values of those traces represent two valuations of the tm_i (τ_i and τ'_i) and two of the spc_i (σ_i and σ'_i).
- ψ_{halt} ensures that there exists a position in the trace assigned to π containing q_f.
- ψ_{init} ensures that all the positions having the tm_i representing 0 contain a B and a q_0, and that positions having the tm_i and spc_i representing 0 contain an h, i.e., the initial configuration is encoded correctly.
- ψ_{cons} checks that all positions of the trace assigned to π in which the valuation of the time and space propositions is the one represented by the traces assigned to the τ_i and σ_i have the same valuation over the other variables. It also checks that there is exactly one letter and one state at each position of the trace assigned to π, and that if two positions in the trace assigned to π have the same time valuations (represented by the trace assigned to τ_i) but not the same space valuations, then at most one of the two satisfies the head proposition h, and they have the same state.

- $\psi_{complete}$ expresses that there exists a position in π matching the tm_i and spc_i with the τ_i and σ_i, i.e., that the complete computation tableau is encoded.
- $\psi_{trans}(q, q', a, a', d)$ checks that the position of the head, the state and the change in the letters are consistent with that transition. For instance, if the σ_i and σ_i' represent the same valuation and the τ_i' represent the successor of the τ_i and the positions in π matching the τ_i and σ_i contain h, then ψ_{trans} ensures that the positions matching τ_i and σ_i contain a and q and the ones matching τ_i' and σ_i' contain a' and q'.

One can then see that this formula is satisfiable if and only if $\mathcal{M}$ accepts in time at most 2^n, as it explicitly describes the existence of an accepting run of $\mathcal{M}$. ◻

We conclude by considering the satisfiability problem for $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$ with arbitrary quantifier prefixes, but restricted to models induced by finite-state systems. The undecidability of satisfiability for arbitrary formulas over finite-state systems can be easily inferred from the proof of undecidability of satisfiability of Finkbeiner and Hahn, as the formulas they construct, if satisfiable, have a finite and ultimately periodic model, which is therefore representable by a finite-state system. For formulas of $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$, we leave decidability open, but prove intractability.

Theorem 8. *The following problem is* TOWER*-hard: Given a HyperLTL*$^1(\mathbf{F}, \mathbf{G})$ *sentence* φ*, does* φ *have a model* $T(\mathcal{K})$ *for some Kripke structure* $\mathcal{K}$*?*

Proof. Let $\mathcal{M}$ be a deterministic Turing machine $(Q, \Sigma, \delta, q_0, q_f)$, let $w \in \Sigma^*$, and let $n = |w|$. We construct a sentence φ that is satisfiable by a Kripke structure if and only if $\mathcal{M}$ accepts w in time $\mathrm{twr}(n)$, where $\mathrm{twr}(0) = 2$ and $\mathrm{twr}(k+1) = 2^{\mathrm{twr}(k)}$ for all $k \in \mathbb{N}$. This problem can easily be shown TOWER-complete under elementary reductions.

To this end, we use the set

$$\mathrm{AP} = \Sigma \cup Q \cup \{\mathrm{rk}_0, \dots, \mathrm{rk}_n\} \cup \{h, 0, 1, 2, in, ni, m_t, m_s, lt, gt\}$$

of propositions. There are two types of traces: Type one traces, marked by the atomic proposition 1, allow us to count up to $\mathrm{twr}(n)$, while type two traces simulate an accepting run of $\mathcal{M}$ on w. Each type two trace encodes one cell of the tape of $\mathcal{M}$ at one step of the computation. We use the formula $\psi_{type} = \forall \pi. 1_\pi \oplus 2_\pi$ to ensure that the traces are split between the two types.

Each type one trace is assigned a *rank* between 0 and n, which is indicated by the propositions $\mathrm{rk}_0, \dots, \mathrm{rk}_n$. A trace of rank $r+1$ encodes a set of traces of rank r. There are two traces of rank zero, which represent zero and one. Thus, we are able to generate $\mathrm{twr}(r)$ traces of rank r for all $0 \leq r \leq n$. For instance, at rank one we have four traces, representing all possible subsets of the two rank zero traces.

Then, we construct an order on those traces and use traces of rank n to simulate the Turing machine. More precisely, each trace of type two is associated to two traces of type one, one for space and one for time, which encode its position on the tape and the step of the computation it describes.

We require that for all type one traces t there is exactly one r such that $\mathrm{rk}_r \in t(0)$, and we then say that t is of rank r. If t is of rank zero then it is either $t_0 = \{1, r_0, 0\}^\omega$ or $t_1 = \{1, r_0\}^\omega$. These properties can easily be expressed in $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$.

A type one trace t of rank $r+1$ is said to contain a type one trace t' of rank r (denoted by $t' \in t$) if and only if there exists $j \in \mathbb{N}$ such that $in \in t'(j)$ and $ni \in t(j)$. To every trace t of rank r we associate a set $S(t)$, which is an element of the set obtained by iterating the power set operator on $\{\{0\}, \{1\}\}$ r times:

- $S(t_0) = \{0\}$ and $S(t_1) = \{1\}$.
- Else, $S(t) = \{S(t') \mid t' \in t\}$.

The following formula is satisfied by $[\pi \to t, \pi' \to t']$ if and only if $t \in t'$:

$$\psi_{in}(\pi, \pi') = 1_\pi \wedge 1_{\pi'} \wedge \mathbf{F}\,(ni_{\pi'} \wedge in_\pi) \wedge \bigvee_{r=0}^{n-1} (\mathrm{rk}_r)_{\pi'} \wedge (\mathrm{rk}_{r+1})_\pi.$$

Furthermore, we define a strict order $<$ over traces of equal rank: $t_0 < t_1$ for the only two traces t_0, t_1 of rank 0, and for all traces t, t' of rank $r+1$, $t < t'$ if and only if there exists t'' of rank r such that

$S(t'') \in S(t') \setminus S(t)$ and for all t^* of rank r such that $t'' < t^*$, $S(t^*) \in S(t)$ if and only if $S(t^*) \in S(t')$. This is defined by analogy with the comparison of two numbers based on their binary representations (assumed to be of the same size). One is larger than the other if there is a position at which it has a 1, the other has a 0, and they are equal on every bit of higher weight. Using this intuition, one can indeed show that $<$ is a strict order.

To express the order $<$ in HyperLTL, we use atomic propositions lt and gt and the following formula to ensure that for all traces t, t' in the model, $t < t'$ if and only if there exists $j \in \mathbb{N}$ such that $lt \in t(j)$ and $gt \in t'(j)$:

$$\forall \pi.\forall \pi'.(1_\pi \wedge 1_{\pi'}) \Rightarrow [\mathbf{F}\,(lt_\pi \wedge gt_{\pi'}) \Leftrightarrow \psi_{0,1} \vee (\psi_{equalrank} \wedge \psi_{order})],$$

where $\psi_{0,1}$ expresses that the value taken by π is t_0 and the one taken by π' is t_1, $\psi_{equalrank}$ expresses that the values of π and π' have the same rank, and

$$\psi_{order} = \exists \tau.\forall \tau'.\psi_{in}(\tau, \pi') \wedge \neg\psi_{in}(\tau, \pi) \wedge \mathbf{F}\,(lt_\tau \wedge gt_{\tau'}) \Rightarrow (\psi_{in}(\tau, \pi) \Leftrightarrow \psi_{in}(\tau', \pi')).$$

We also require that two traces t, t' of equal rank but incomparable for $<$ (i.e. $S(t) = S(t')$) must be equal. This is expressed by the formula

$$\forall \pi.\forall \pi'.(1_\pi \wedge 1_{\pi'}) \Rightarrow \Big[\mathbf{F}\,(lt_\pi \wedge gt_{\pi'}) \vee \mathbf{F}\,(gt_\pi \wedge lt_{\pi'}) \vee \mathbf{G}\,\big(\bigwedge_{a \in \mathrm{AP}} a_\pi \Leftrightarrow a_{\pi'}\big)\Big].$$

We then define a successor relation sc on traces of equal rank. We have $sc(t_0, t_1)$ and not $sc(t_1, t_0)$, $sc(t_1, t_1)$, or $sc(t_1, t_1)$. For all traces t, t' of rank $r > 0$, $sc(t, t')$ expresses that there exists a trace t'' of rank $r - 1$ such that, $S(t'') \in S(t') \setminus S(t)$ and for all $t^* < t''$, $S(t^*) \in S(t) \setminus S(t')$. Intuitively, this is analogous to the definition of the successor in binary: In order to add one to a binary number, one has to find the bit of least weight such that this bit is 0 and all the bits of lower weights are 1. Then one has to turn this bit to one and all the lower ones to 0. The following formula is satisfied by $[\pi \to t, \pi' \to t']$ if and only if $sc(t, t')$ holds:

$$\psi_{sc}(\pi, \pi') = \exists \tau.\forall \tau'.(1_\pi \wedge 1_{\pi'}) \wedge \Big[1_{\tau'}) \Rightarrow$$
$$1_\tau \wedge \psi_{in}(\tau, \pi') \wedge \neg\psi_{in}(\tau, \pi) \wedge (\mathbf{F}\,(lt_{\tau'} \wedge gt_\tau) \Rightarrow \psi_{in}(\tau', \pi) \wedge \neg\psi_{in}(\tau', \pi'))\Big].$$

We then ensure that all possible $S(t)$ are generated: We state the existence of a trace t_0^r for each $1 \leq r \leq n$ such that for all $j \in \mathbb{N}$, $ni \notin t_0^r(j)$. The trace t_0^r represents the empty set at rank r. The following formula expresses the existence of those traces:

$$\exists \tau_0^1 \ldots \exists \tau_0^n. \bigwedge_{r=1}^{n} 1_{\tau_0^r} \wedge (\mathrm{rk}_r)_{\tau_0^r} \wedge \mathbf{G}\,(\neg(ni)_{\tau_0^r}).$$

We also state that for all r and for all traces t of rank r, if t is not maximal for $<$ (i.e., if it does not contain every trace of rank $r - 1$), then there exists a trace t' such that $sc(t, t')$:

$$\forall \pi.\exists \tau.1_\pi \Rightarrow \psi_{sc}(\pi, \tau) \vee \psi_{max}(\pi)$$

where

$$\psi_{max}(\pi) = \forall \pi'.[1_{\pi'} \wedge \bigvee_{r=0}^{n-1} \mathrm{rk}_{r\pi'} \wedge \mathrm{rk}_{r+1\pi}] \Rightarrow \psi_{in}(\pi', \pi).$$

As a result, we have ensured that all the $S(t)$ were generated. In particular, we have $\mathrm{twr}(n)$ traces of rank n, and we have a successor relation sc. We can use these traces as a counter from 0 to $\mathrm{twr}(n) - 1$.

Finally, for convenience, we ensure that S is injective, i.e., for all traces t, t' of type one, if $S(t) = S(t')$ (i.e., if t and t' have the same rank but are incomparable for $<$) then $t = t'$. The following formula expresses it:

$$\forall \pi.\forall \pi'.\Big[\big(\bigvee_{r=0}^{n} (\mathrm{rk}_r)_\pi \wedge (\mathrm{rk}_r)_{\pi'}\big) \wedge \neg\mathbf{F}\,((lt_\pi \wedge gt_{\pi'}) \vee (lt_{\pi'} \wedge gt_\pi))\Big] \Rightarrow \mathbf{G}\,\big(\bigwedge_{a \in \mathrm{AP}} a_\pi \Leftrightarrow a_{\pi'}\big)$$

We associate to every pair (t, t') of traces of type one and rank n (t for time and t' for space) a unique type two trace representing the content of the tape at the position represented by t_s at the time represented by t_t. We use the atomic propositions m_t and m_s to encode this relation. The following formulas express it:

$$\forall \pi_t . \forall \pi_s . \exists \tau . (1_{\pi_t} \wedge 1_{\pi_s} \wedge (\mathrm{rk}_n)_{\pi_t} \wedge (\mathrm{rk}_n)_{\pi_s}) \Rightarrow \mathbf{F}\,((m_s)_{\pi_s} \wedge (m_s)_\tau) \wedge \mathbf{F}\,((m_t)_{\pi_t} \wedge (m_t)_\tau) \wedge 2_\tau$$

expresses that to each pair (t, t') of rank n traces we associate a type two trace as explained above, and

$$\forall \pi . \forall \pi' . \forall \tau . \Big[\mathbf{F}\,((m_s)_\pi \wedge (m_s)_\tau) \wedge \mathbf{F}\,((m_s)_{\pi'} \wedge (m_s)_\tau) \wedge \mathbf{F}\,((m_t)_\pi \wedge (m_t)_\tau) \wedge \mathbf{F}\,((m_t)_{\pi'} \wedge (m_t)_\tau)\Big]$$
$$\Rightarrow \mathbf{G}\,\Big(\bigwedge_{a \in \mathrm{AP}} a_\pi \Leftrightarrow a_{\pi'} \Big)$$

expresses that this type two trace is unique.

We can now define a formula $\psi_{sametime}(\tau, \tau')$ satisfied in the model by $[\tau \to t, \tau' \to t']$ if and only if t and t' are of type two and are associated to the same step of the computation:

$$\psi_{sametime}(\tau, \tau') = \exists \pi_t . 2_\tau \wedge 2_{\tau'} \wedge \mathbf{F}\,((m_t)_{\pi_t} \wedge (m_t)_\tau) \wedge \mathbf{F}\,((m_t)_{\pi_t} \wedge (m_t)_{\tau'})$$

We also define a formula $\psi_{nexttime}(\tau, \tau')$ satisfied in the model by $[\tau \to t, \tau' \to t']$ if and only if t and t' are of type two and the time associated to t' is the successor of the time associated to t:

$$\psi_{nexttime}(\tau, \tau') = \exists \pi_t . \exists \pi'_t . 2_\tau \wedge 2_{\tau'} \wedge \psi_{next}(\pi_t, \pi'_t) \wedge \mathbf{F}\,((m_t)_{\pi_t} \wedge (m_t)_\tau) \wedge \mathbf{F}\,((m_t)_{\pi'_t} \wedge (m_t)_{\tau'})$$

We define $\psi_{nextspace}$ similarly.

With these formulas at hand, we can now simulate the run of $\mathcal{M}$: The consistency and the transitions of the machine can easily be implemented using the *sc* and $<$ relations. For instance, the following formula expresses that if at some step of the run the head does not point to a position, then the letter at that position should not change between this step and the next one:

$$\forall \tau . \forall \tau' . \psi_{nexttime}(\tau, \tau') \wedge \psi_{samespace}(\tau, \tau') \wedge \neg h_\tau \Rightarrow \bigwedge_{a \in \Sigma} (a_\tau \Rightarrow a_{\tau'}).$$

If the final formula has a model, then $\mathcal{M}$ accepts w , and then we can construct a model satisfying the formula which is finite (as there are finitely many type one and type two traces) and in which all traces are ultimately periodic (as there are finitely many of them, we only need finitely many positions to encode all the relations). Thus, this model can be represented by a Kripke structure of nonelementary size.

Conversely, if the formula has a model represented by a Kripke structure, then in particular it is satisfiable. Thus, $\mathcal{M}$ accepts w (one can construct an accepting run of $\mathcal{M}$ from a model of the formula).

Furthermore, even though the construction described in this proof may yield formulas of alternation depth greater than one, we can reduce the depth to one using Lemma 2: If the final formula has a model, then it has a finite and ultimately periodic one. The construction of Lemma 2 encodes a finite number of new relations over the traces. Hence, the resulting $\forall^*\exists^*$ formula also has a finite and ultimately periodic model, which is therefore also the set of traces of a Kripke structure. □

Let us conclude by remarking that the satisfiability problem for $\mathrm{HyperLTL}^1(\mathbf{F}, \mathbf{G})$ over Kripke structures is different from the general one, i.e., there are satisfiable formulas which are not satisfied by the set of traces of any Kripke structure. Consider for instance the sentence $\forall \pi . \exists \pi' . \mathbf{G}\,(a_\pi \Rightarrow a_{\pi'}) \wedge \mathbf{F}\,(\neg a_\pi \wedge a_{\pi'})$. It is satisfied by $\{a\}^* \emptyset^\omega$.

Suppose there exists a Kripke structure $\mathcal{K}$ with a set of traces satisfying this sentence. We define inductively an increasing sequence of finite trace prefixes p_n for $n \in \mathbb{N}$ as $p_0 = \varepsilon$ and $p_{n+1} = p_n\{a\}$ if $p_n\{a\}$ is a prefix of a trace of $\mathcal{K}$, and $p_{n+1} = p_n\emptyset$ otherwise. As the set of traces of a Kripke structure is closed, the limit t of this sequence is a trace of $\mathcal{K}$. As $\mathcal{K}$ satisfies the sentence, there exists t' such that for all j if $a \in t(j)$ then $a \in t'(j)$ and there exists j^* such that $a \in t'(j^*)$ and $a \notin t(j^*)$. In particular, there exists a minimal such j^*. Then $p_{j^*+1} = p_{j^*}\emptyset$, but $p_{j^*}\{a\}$ is a prefix of t', thereby contradicting the choice of p_{j^*+1}, as we prefer to extend by $\{a\}$ instead of $\emptyset$. Thus, this satisfiable sentence is not satisfiable by the set of traces of a finite Kripke structure.

5 Conclusion

We have shown that HyperLTL satisfiability can be decidable, either if one restricts the space of models one is interested in to sufficiently simple ones, or if one restricts the alternation and temporal depth of the formulas under consideration. In particular, we have investigated the formulas of temporal depth one without untils. An interesting open problem is to extend the decidability result presented in Theorem 6 to formulas with untils. Also, we claimed no lower bound on the problem solved in Theorem 6. We claim there is an ExpSpace lower bound obtained by encoding exponential space Turing machines, but the exact complexity of the problem is left open. Another interesting problem left open is the decidability of HyperLTL$^1(\mathbf{F}, \mathbf{G})$ over Kripke structures. We have presented a Tower lower bound in Theorem 8, but it is open whether the problem is indeed decidable.

In general, restricting the space of models turns out to be more fruitful than to restrict the formulas under consideration, as satisfiability is undecidable for extremely simple formulas (simplicity being measured in alternation depth and temporal depth). An interesting challenge pertains to finding other measures of simplicity that yield larger decidable fragments.

References

1. Bonakdarpour, B., Finkbeiner, B.: Runtime verification for HyperLTL. In: Falcone, Y., Sánchez, C. (eds.) RV 2016. LNCS, vol. 10012, pp. 41–45. Springer (2016)
2. Clarkson, M.R., Finkbeiner, B., Koleini, M., Micinski, K.K., Rabe, M.N., Sánchez, C.: Temporal logics for hyperproperties. In: Abadi, M., Kremer, S. (eds.) POST 2014. LNCS, vol. 8414, pp. 265–284. Springer (2014)
3. Clarkson, M.R., Schneider, F.B.: Hyperproperties. Journal of Computer Security 18(6), 1157–1210 (2010)
4. Coenen, N., Finkbeiner, B., Hahn, C., Hofmann, J.: The hierarchy of hyperlogics (2019), to appear at LICS 2019.
5. Coenen, N., Finkbeiner, B., Sanchez, C., Tentrup, L.: Verifying hyperliveness (2019), to appear at CAV 2019.
6. Demri, S., Schnoebelen, P.: The complexity of propositional linear temporal logics in simple cases. Inf. Comput. 174(1), 84–103 (2002)
7. Finkbeiner, B., Hahn, C.: Deciding hyperproperties. In: Desharnais, J., Jagadeesan, R. (eds.) CONCUR 2016. LIPIcs, vol. 59, pp. 13:1–13:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2016)
8. Finkbeiner, B., Hahn, C., Lukert, P., Stenger, M., Tentrup, L.: Synthesizing reactive systems from hyperproperties. In: Chockler, H., Weissenbacher, G. (eds.) CAV 2018 (Part I). LNCS, vol. 10981, pp. 289–306. Springer (2018)
9. Finkbeiner, B., Hahn, C., Stenger, M.: EAHyper: Satisfiability, implication, and equivalence checking of hyperproperties. In: Majumdar, R., Kuncak, V. (eds.) CAV 2017 (Part II). LNCS, vol. 10427, pp. 564–570. Springer (2017)
10. Finkbeiner, B., Hahn, C., Stenger, M., Tentrup, L.: RVHyper: A runtime verification tool for temporal hyperproperties. In: Beyer, D., Huisman, M. (eds.) TACAS 2018 (Part II). LNCS, vol. 10806, pp. 194–200. Springer (2018)
11. Finkbeiner, B., Rabe, M.N., Sánchez, C.: Algorithms for model checking HyperLTL and HyperCTL*. In: Kroening, D., Pasareanu, C.S. (eds.) CAV 2015 (Part I). LNCS, vol. 9206, pp. 30–48. Springer (2015)
12. Finkbeiner, B., Zimmermann, M.: The first-order logic of hyperproperties. In: Vollmer, H., Vallée, B. (eds.) STACS 2017. LIPIcs, vol. 66, pp. 30:1–30:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017)
13. Goguen, J.A., Meseguer, J.: Security policies and security models. In: 1982 IEEE Symposium on Security and Privacy. pp. 11–20. IEEE Computer Society (1982)
14. Halpern, J.Y., Harper, R., Immerman, N., Kolaitis, P.G., Vardi, M.Y., Vianu, V.: On the unusual effectiveness of logic in computer science. Bulletin of Symbolic Logic 7(2), 213–236 (2001)
15. Hopcroft, J.E., Ullman, J.D.: Formal languages and their relation to automata. Addison-Wesley series in computer science and information processing, Addison-Wesley (1969)
16. Kurshan, R.P.: Transfer of model checking to industrial practice. In: Clarke, E.M., Henzinger, T.A., Veith, H., Bloem, R. (eds.) Handbook of Model Checking., pp. 763–793. Springer (2018)
17. McCullough, D.: Noninterference and the composability of security properties. In: 1988 IEEE Symposium on Security and Privacy. pp. 177–186. IEEE Computer Society (1988)
18. McCullough, D.: A hookup theorem for multilevel security. IEEE Trans. Software Eng. 16(6), 563–568 (1990)
19. McLean, J.: A general theory of composition for trace sets closed under selective interleaving functions. In: 1994 IEEE Computer Society Symposium on Research in Security and Privacy. pp. 79–93. IEEE Computer Society (1994)
20. Millen, J.K.: Unwinding forward correctability. Journal of Computer Security 3(1), 35–54 (1995)
21. Pnueli, A.: The Temporal Logic of Programs. In: FOCS 1977. pp. 46–57 (1977)

22. Schmitz, S.: Complexity hierarchies beyond elementary. TOCT 8(1), 3:1–3:36 (2016)
23. Sistla, A.P., Clarke, E.M.: The complexity of propositional linear temporal logics. Journal of the ACM 32(3), 733–749 (1985)
24. Stockmeyer, L.J., Meyer, A.R.: Word problems requiring exponential time: Preliminary report. In: Aho, A.V., Borodin, A., Constable, R.L., Floyd, R.W., Harrison, M.A., Karp, R.M., Strong, H.R. (eds.) STOC 1973. pp. 1–9. ACM (1973)
25. Zdancewic, S., Myers, A.C.: Observational determinism for concurrent program security. In: CSFW 2003. p. 29. IEEE Computer Society (2003)